FM-3 Programming Module Reference Manual

P/N 400508-01
Revision: A2
Date: August 23, 2000
© EMERSON Motion Control, Inc. 2000

FM-3
Programming Module
Reference Manual

Information furnished by EMERSON Motion Control is believed to be accurate and reliable. However, no responsibility is assumed by EMERSON Motion Control for its use. EMERSON Motion Control reserves the right to change the design or operation of the equipment described herein and any associated motion products without notice. EMERSON Motion Control also assumes no responsibility for any errors that may appear in this document. Information in this document is subject to change without notice.

P/N 400508-01
Revision: A2
Date: August 23, 2000
© EMERSON Motion Control™, Inc. 2000

© EMERSON Motion Control, Inc. 2000

Part Number: 400508-01

Revision: A2

Date: August 2000

Printed in United States of America

Information in this document is subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of EMERSON Motion Control.

The following are trademarks of EMERSON Motion Control and may not be reproduced in any fashion without written approval of EMERSON Motion Control: EMERSON Motion Control, EMERSON Motion Control PowerTools.

EMERSON Motion Control, a Division of Emerson Electric Co.

EMERSON Motion Control, Inc. is not affiliated with Microsoft Corporation, owner of the MicroSoft, Windows, and Windows NT trademarks.

This document has been prepared to conform to the current released version of the product. Because of our extensive development efforts and our desire to further improve and enhance the product, inconsistencies may exist between the product and documentation in some instances. Call your customer support representative if you encounter an inconsistency.

Customer Service

It is EMERSON Motion Control's goal to ensure your greatest possible satisfaction with the operation of our products. We are dedicated to providing fast, friendly, and accurate assistance. That is why we offer you so many ways to get the support you need. Whether by phone, fax or modem, you can access EMERSON Motion Control support information 24 hours a day, seven days a week.

EMERSON Motion Control
12005 Technology Drive
Eden Prairie, Minnesota 55344
U.S.A.

FAX **(952)995-8011**

You can FAX questions and comments to EMERSON Motion Control. Just send a FAX to the number listed above.

Website: **www.emersondrivesolutions.com**

If you have Internet capabilities, you also have access to technical support using our Website at http://www.emersondrivesolutions.com. The Website includes technical notes, frequently asked questions, release notes and other technical documentation. This direct technical support connection lets you request assistance and exchange software files electronically.

Service Support **(952)995-8033**

EMERSON Motion Control's products are backed by a team of professionals who will service your installation wherever it may be. Our customer service center in Eden Prairie, Minnesota is ready to help you solve those occasional problems over the telephone. Our customer service center is available 24 hours a day for emergency service to help speed any problem solving. Also, all hardware replacement parts, should they ever be needed, are available through our customer service organization.

When you call, please be at your computer, have your documentation in hand, and be prepared to provide the following information:

Product version number, found by choosing **About** from the **Help** menu.

The type of controller or product you are using.

Exact wording of any messages that appear on your screen.

What you were doing when the problem occurred.

How you tried to solve the problem.

You can also contact us by email: service@emersondrivesolutions.com

Need on-site help? EMERSON Motion Control provides service, in most cases, the next day. Just call EMERSON's customer service center when on-site service or maintenance is required.

Training Services **(800)397-3786 or Fax (952)995-8011**

EMERSON Motion Control maintains a highly trained staff of instructors to familiarize customers with EMERSON Motion Control's products and their applications. A number of courses are offered, many of which can be taught in your plant upon request.

Application Engineering **(800)893-2321**

An experienced staff of factory application engineers provides complete customer support for tough or complex applications. Our engineers offer you a broad base of experience and knowledge of electronic motion control applications. Application engineers may be contacted by email at applengr@emersondrivesolutions.com or at the telephone number listed above.

Sales: **(952)995-8000**

Email: sales@emersondrivesolutions.com
Authorized EMERSON Motion Control distributors may place orders directly with our Order Processing department by calling the number listed above. For information on your local distributor, call EMERSON Motion Control.

Document Conventions

Manual conventions have been established to help you learn to use this manual quickly and easily. As much as possible, these conventions correspond to those found in other Microsoft® Windows®[1] documentation.

Menu names and options are printed in bold type: the **File** menu.

Dialog box names begin with uppercase letters: the Axis Limits dialog box.

Dialog box field names are in quotes: “Field Name”.

Button names are in italic: *OK* button.

Source code is printed in Courier font: `Case ERMS.`

In addition, you will find the following typographic conventions throughout this manual.

This	Represents
bold	Characters that you must type exactly as they appear. For example, if you are directed to type **a:setup**, you should type all the bold characters exactly as they are printed.
italic	Place holders for information you must provide. For example, if you are directed to type *filename*, you should type the actual name for a file instead of the word shown in italic type.
ALL CAPITALS	Directory names, file names, key names, and acronyms.
SMALL CAPS	Non-printable ASCII control characters.
KEY1+KEY2 example: (Alt+F)	A plus sign (+) between key names means to press and hold down the first key while you press the second key.
KEY1,KEY2 example: (Alt,F)	A comma (,) between key names means to press and release the keys one after the other.

Special nomenclature is provided for people working with these products, as follows:

Warning

For the purpose of this manual and product, “Warning” indicates death, severe personal injury or substantial property damage CAN result if proper precautions are not taken.

Caution

For the purpose of this manual and product, “Caution” indicates minor personal injury or property damage CAN result if proper precautions are not taken.

Note

For the purpose of this manual and product, “Note” indicates information about the product or the respective part of the manual which is essential to highlight.

Throughout this manual, the word “drive” refers to an E Series drive.

1.See page ii of the preface.

Safety Instructions

General Warning

Failure to follow safe installation guidelines can cause death or serious injury. The voltages used in the product can cause severe electric shock and/or burns, and could be lethal. Extreme care is necessary at all times when working with or adjacent to it. The installation must comply with all relevant safety legislation in the country of use.

Qualified Person

For the purpose of this manual and product, a “qualified person” is one who is familiar with the installation, construction and operation of the equipment and the hazards involved. In addition, this individual has the following qualifications:

- Is trained and authorized to energize, de-energize, clear, ground and tag circuits and equipment in accordance with established safety practices.
- Is trained in the proper care and use of protective equipment in accordance with established safety practices.
- Is trained in rendering first aid.

Reference Materials

The following related reference and installation manuals may be useful with your particular system.

- *FM-3 Speed Module Installation Manual* (P/N 400508-02)
- *Epsilon Eb and E Series EN Drives Reference Manual* (P/N 400501-01)
- *PowerTools Software User’s Guide* (P/N 400503-01)

Safety Considerations

Safety Precautions

This product is intended for professional integration into a complete system. If you install the product incorrectly, it may present a safety hazard. The product and system may use high voltages and currents, carry a high level of stored electrical energy, or control mechanical equipment that can cause injury.

You should give close attention to the electrical installation and system design to avoid hazards either in normal operation or in the event of equipment malfunction. System design, installation, commissioning and maintenance must be carried out by personnel who have the necessary training and experience. Read and follow this safety information and the instruction manual carefully.

Enclosure

This product is intended to be mounted in an enclosure which prevents access except by trained and authorized personnel, and which prevents the ingress of contamination. This product is designed for use in an environment classified as pollution degree 2 in accordance with IEC664-1. This means that only dry, non-conducting contamination is acceptable.

Setup, Commissioning and Maintenance

It is essential that you give careful consideration to changes to drive settings. Depending on the application, a change could have an impact on safety. You must take appropriate precautions against inadvertent changes or tampering. Restoring default parameters in certain applications may cause unpredictable or hazardous operation.

Safety of Machinery

Within the European Union all machinery with which this product is used must comply with Directive 89/392/EEC, Safety of Machinery.

The product has been designed and tested to a high standard, and failures are very unlikely. However the level of integrity offered by the product's control function – for example stop/start, forward/reverse and maximum speed – is not sufficient for use in safety-critical applications without additional independent channels of protection. All applications where malfunction could cause injury or loss of life must be subject to a risk assessment, and further protection provided where needed.

Warning

General warning
Failure to follow safe installation guidelines can cause death or serious injury. The voltages used in this unit can cause severe electric shock and/or burns, and could be lethal. Extreme care is necessary at all times when working with or adjacent to this equipment. The installation must comply with all

relevant safety legislation in the country of use.

AC supply isolation device
The AC supply must be removed from the drive using an approved isolation device or disconnect before any servicing work is performed, other than adjustments to the settings or parameters specified in the manual. The drive contains capacitors which remain charged to a potentially lethal voltage after the supply has been removed. Allow at least 3 minutes for Epsilon and 30 seconds for E Series drives after removing the supply before carrying out any work which may involve contact with electrical connections to the drive.

Products connected by plug and socket
A special hazard may exist where the drive is incorporated into a product which is connected to the AC supply by a plug and socket. When unplugged, the pins of the plug may be connected to the drive input, which is only separated from the charge stored in the bus capacitor by semiconductor devices. To avoid any possibility of electric shock from the pins, if they are accessible, a means must be provided for automatically disconnecting the plug from the drive (that is, a latching contactor).

Grounding (Earthing, equipotential bonding)
The drive must be grounded by a conductor sufficient to carry all possible fault current in the event of a fault. The ground connections shown in the manual must be followed.

Fuses
Fuses or over-current protection must be provided at the input in accordance with the instructions in the manual.

Isolation of control circuits
The installer must ensure that the external control circuits are isolated from human contact by at least one layer of insulation rated for use at the applied AC supply voltage.

Table of Contents

Customer Service iii
Reference Materials vi
Safety Considerations vii

Introduction 1

Operational Overview 3

Software Interface 3
PowerTools FM-3 Setup Software 3
Keypad Interface 4
How Motion Works 6
How Jogging Works 6
How Home Works 7
How Indexes Work 17
How Communications Work 22

Setting Up Parameters 25

Setup View 25
Status Online Tab (Online Only) 27
User Units View 30
Master Units View 34
Position View 37
Velocity View 41
Ramps View 43
Torque View 45
Tuning View 47
Faults View 50
PLS View 51
I/O Setup Group 54
Selector View 54
Assignments 56
Assignments View 57
Input Lines View 60
Output Lines View 61
Motion Group 62
Jog View 62
Home View 68

Index View 72
Network Group 79
Modbus View 79

Programming 81

Parameter Descriptions 99
Parameters Saved to NVM 127

Installation 129
Basic Installation Notes 129
Mechanical Installation 129
Software Installation 131
PC System Configuration 131
Preparing for PowerTools FM-3 Installation 131
Installing PowerTools FM-3 131
Starting and Exiting PowerTools 132
Accessing Help 133

Quick Start 135
Basic Setup Steps 135
Example Application Start Up 149
. 156

Tuning Procedures 157
PID vs. State-Space 157
Tuning Procedure 158
Tuning Parameters 160
Determining Tuning Parameter Values 162

Diagnostics and Troubleshooting 169
Diagnostic Display 169
Fault Codes 170
Brake Operation 176
Analog Outputs 177
Diagnostic Analog Output Test Points 178
Drive Faults 180
Error Messages 180
Programming Error Messages 182

Online Status Indicators 187

Specifications **189**

Dimensions and Clearances 189
Cable Diagrams 191

Glossary **205**

Index **213**

Introduction

FM-3 is a compact and rugged function module that attaches to the front of the E Series drive. It provides eight digital input lines and four digital output lines, in addition to the four input and three output lines available on the E Series EN drive.

Unlike other function modules, the FM-3 offers complex motion profiling. A complex motion profile consists of two or more indexes that are executed in sequence such that the final velocity of each index except the last is non-zero. Logical instructions between index statements can provide a powerful tool for altering motion profiles 'on the fly'. The FM-3 defines complex motion by a configuration file that includes setups, function assignments and programs. The configuration file is created using EMERSON Motion Control PowerTools® FM-3 software[1]. Setup views have the same look and feel as dialog boxes. The wiring of input and output functions is done through assignments in the software. Emerson Motion Control PowerTools FM-3 is an easy-to-use Microsoft® Windows®[2] based setup and diagnostics tool.

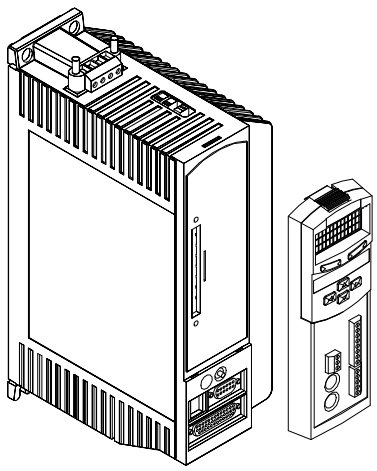


Figure 1: E Series EN Drive with FM-3 Function Module

Note that the E Series drive's firmware is disabled whenever a Function Module such as the FM-3 is attached. Therefore, if the E Series drive's hardware is FM compatible, then the E Series drive's firmware can be any version because the programming features reside in the function module's flash memory. Flash files used for firmware upgrades are available on the Emerson Motion Control webpage.

1.In this manual, Emerson Motion Control PowerTools FM-3 software will be referred to as PowerTools FM-3.

2. See page ii of the preface

The FM-3 stores E Series drive setup parameters within the module itself. This allows you to transfer the FM-3 to another E Series drive without losing setup parameters.

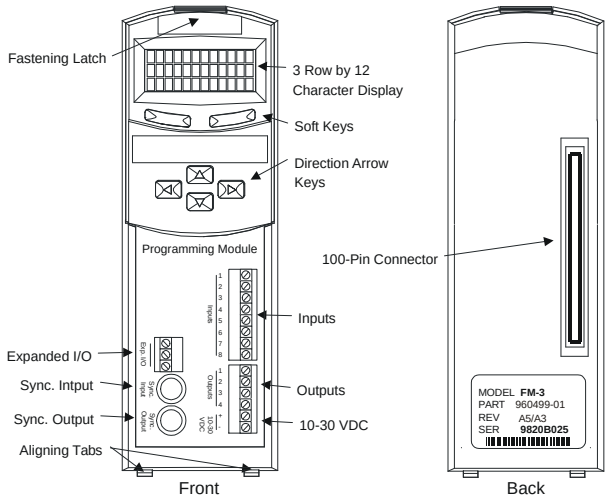


Figure 2: FM-3 Programming Module Features

Operational Overview

This section provides a complete functional description of the FM-3. It is intended to provide you with a thorough understanding of all operations. The description includes references to many FM-3 and E Series drive parameters which can be displayed and/or edited using Emerson Motion Control PowerTools® FM-3 software, or through any Modbus interface.

The FM-3 augments the E Series drive by providing the ability to implement programs written using PowerTools FM-3. When an FM-3 is attached to an E Series drive, it overrides the operation and user accessible features of the drive. The drive's basic operating modes (Pulse, Velocity and Torque) are not available when an FM-3 is attached.

The FM-3 stores E Series drive setup parameters within the module itself. This allows you to transfer the FM-3 to another E Series drive without losing setup parameters.

The FM-3 allows you to set up 55 different indexes, Jog functions and multiple Homes. It also provides eight digital input lines and four digital output lines in addition to the four input and three output lines available on the drive.

Software Interface

The FM-3 is set up using PowerTools FM-3 software. PowerTools FM-3 is an easy-to-use Windows® based setup and diagnostics tool. It provides you with the ability to create, edit and maintain your drive's setup. You can download or upload your setup data to or from a device. You can also save it to a file on your PC or print it for review or permanent storage.

PowerTools FM-3 Setup Software

PowerTools FM-3 is designed to be the easiest to use software available for single axis motion controllers.

Features

- "Hierarchy View" for quick navigation to any setup screen
- Simple I/O function assignments
- Powerful online diagnostic capabilities
- Programming

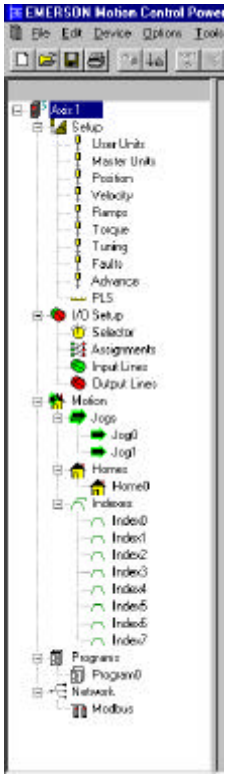


Figure 3: Hierarchy View

The “Hierarchy View” (shown above) contains expandable groups of parameters. The groups can be expanded and contracted just like folders in Windows Explorer. Left clicking on a view name in the Hierarchy view displays that View on the right side of the computer screen.

To setup a drive the user simply steps through the Hierarchy View from top to bottom. Simple applications can be setup in a matter of minutes. Many of the views have Assignment tabs that display the assignments pertaining to the view (i.e. on the Position view, the position assignments are displayed).

Keypad Interface

The keypad on the front of the FM-3 provides navigation through a menu of common parameters and displays of current functions. Navigation through the menu is accomplished with the six keys located below the display. The top two keys are called the "soft keys" because they relate to the commands located directly above each key on the LCD. These keys are used to select the operation (e.g. Modify, Ok, Cancel), parameter group, and/or to validate

information. The four arrow keys are used to navigate through parameter groups, select a specific parameter to be modified, and to modify digital and numeric data.

The operation of the arrow keys is dependent upon the type of parameter which is being modified.

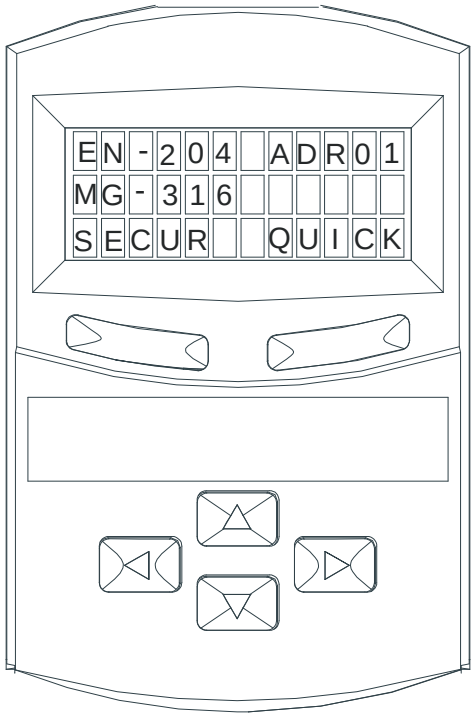


Figure 4: FM-3 Display Screen and Keypad

On the Menu screen, the drive type and axis address are always shown on the top line of the display. The second line shows the motor type. If a user defined motor is selected, the user defined motor name will appear. The third line shows two parameter group names, one above each of the soft keys.

From the Menu screen, the user selects a group of drive parameters to work with. The group names are scrolled using the left/right direction keys. The groups correspond roughly to the tabs used by the PowerTools software. The groups are shown cyclically and wrap around.

The E Series drive parameters available with the FM-3 keypad are arranged into seven groups (see list below). Upon power-up the FM-3 will display the default parameter groups “SECUR” (left soft key) and “QUICK” (right soft key).

- QUICK (Quick)
- PROG (Program)

- INDEX (Index)
- POSN (Position)
- HOME (Home)
- JOG (Jog)
- RAMPS (Ramps)
- SECUR (Security)

Parameter Screens

After selecting a group using one of the soft keys, the FM-3 will display a Parameter screen for that group. This screen could be either the first screen in the group or the last screen you used in that group. The FM-3 keeps track of the last Parameter screen viewed in each group and returns to that screen when you come back to the group. This is reset on power-up and the FM-3 displays the first Parameter screen in the group.

In this screen, the parameter name is shown on the first line of the display. The up/down arrow keys are used to scroll through the parameters available in the selected group.The second line displays the condition or value of parameters. The third line displays the soft key actions.

The left/right arrow keys are used to scroll through the parameters when the “<“ and “>” symbols are shown.

Numeric parameter units are sometimes shown before the actual value, because the parameter value and the units cannot be displayed on one line. The unit of measure will appear on the second line for about one second. Then the actual parameter value will appear. The parameter value is updated about five times a second.

How Motion Works

The FM-3 provides three types of motion: jogging, homes and indexing. Only one index, jog or home may be in process at any given moment (exclusionary motion types). Through assignments and programs, the FM-3 can sequentially run various motion routines. The Positive direction parameter affects all motion types by specifying which direction of motor revolution (CW or CCW) is considered motion in the “+” direction.

How Jogging Works

Jogging produces rotation of the motor at controlled velocities in a positive or negative direction.

Assignments to jogs are level sensitive such that when the jog input is turned on, jogging begins and continues jogging until the jog input is removed.

Each jog has its own acceleration and deceleration ramp along with a specified velocity. Jogging has no distance parameter associated with it. If trying to move a specific distance or to a known position, then an index is used.

Jogging in the opposite direction will move off a travel limit (use Jog + to move off a Travel Limit -).

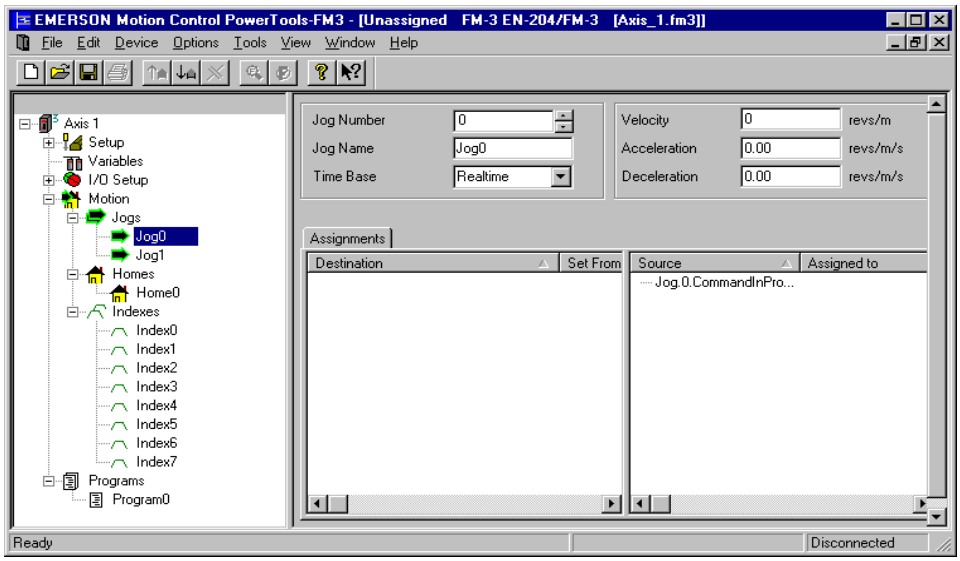


Figure 5: Jog Tab

How Home Works

The Home is used in applications in which the axis must be precisely aligned with some part of the machine. The Home is initiated in one of three ways: with the Initiate Destination function found in the Assignments view, through a program, or with the Online tab. A Home or Define Home is required to set the Absolute Position Valid so that any index to absolute position can work.

The FM-3 can home the motor to an external sensor, the motor's encoder marker pulse, or to a sensor and then to the encoder marker pulse.

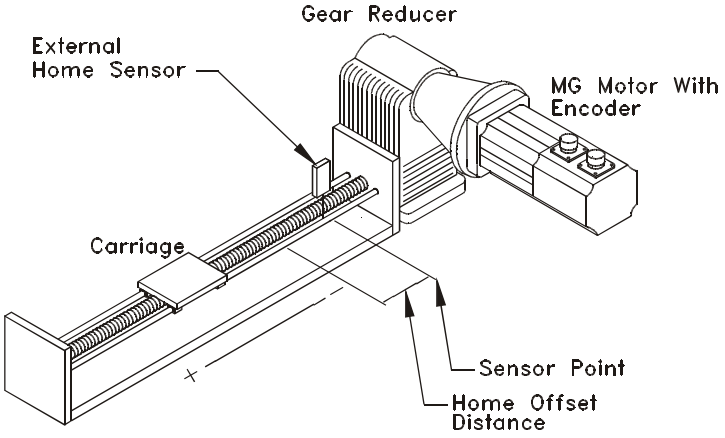


Figure 6: Basic Home Function, Example

The figure above show a basic home function using a ball screw. This example uses most of the setup features in the PowerTools FM-3 Home tab.

Home Sequence

1. Back off the sensor (if on the sensor. This step is optional).
2. Move to the external home sensor to establish a home reference point.
3. Next it will move to the Offset position.
4. Then the command and feedback positions are set to the value entered into the End of Home Position.

Homing to the motor's encoder marker will establish the most accurate and repeatable home position. This method will position the motor relative to the location of the rising edge of the encoder marker pulse. Most applications will use a sensor and marker to find an accurate home position in the vicinity of the home sensor.

Several parameters affect how the Home function operates. Each of these parameters are explained in detail on the following pages.

Note

The Home function will NOT be initiated when any other motion command is in progress.

Establishing a Home Reference Position

The first step in setting up a home is to select the desired home reference type. The Home Reference parameter selected determines how the Home Reference Position is established.

PowerTools FM-3 allows selection of one of three different Home References: Sensor, Marker, or Sensor and Marker.

Sensor

Selecting Sensor means the rising edge of the Home Sensor input function is used to establish the home reference position.

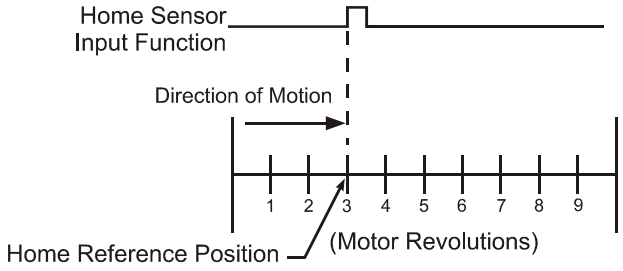


Figure 7: Sensor Home Reference Position

Marker

Selecting Marker means the rising edge of the motor's encoder marker channel is used to establish the reference position.

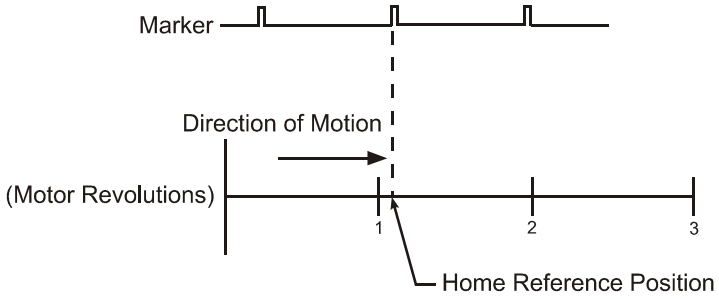


Figure 8: Marker Home Reference Position

Sensor and Marker

Selecting Sensor and Marker means the reference position is established using the first marker rising edge after the device sees the rising edge of the Home Sensor input function.

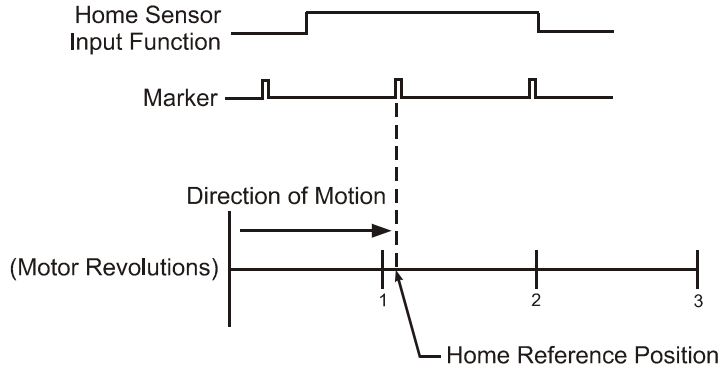


Figure 9: Sensor and Marker Home Reference Position Example 1

Accuracy and Repeatability

The amount of accuracy your application requires will determine the Home Reference option you select. Homing to an external sensor only will establish a repeatable home position within 0.08 revolutions at 3000 RPMs (1.6 msec sensor capture interval).

Note

The data above assumes the use of a perfectly repeatable home sensor.

In Sensor and Marker applications, the marker must be at least 1.6msec after the rising edge of the sensor input to be considered a valid marker pulse.

Note

At 1000 RPM, the motor will travel 0.0266 revolutions (or 9.6°) in 1.6msec.

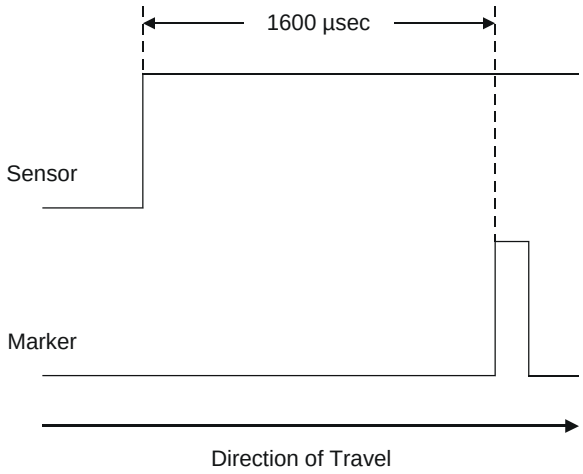


Figure 10: Sensor and Marker Home Reference Position Example 2

The Home Sensor must be "On" for at least 1.6 msec to guarantee that it will be recognized.

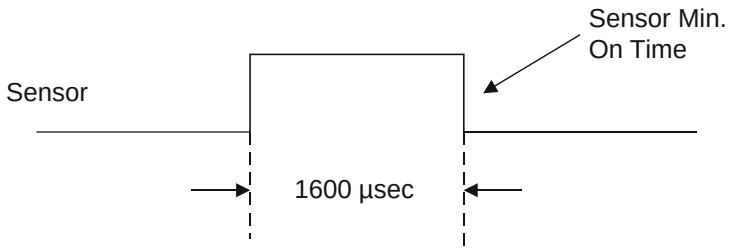


Figure 11: Sensor and Marker Home Reference Position Example 3

Home Offset

The Home Offset is the distance from the Reference Position to the final stopping point at the end of the homing sequence. Regardless of the value you enter for the Offset or which Home Reference you choose, there is always an offset inherent in the homing process.

The user may either specify a desired offset or allow the drive to calculate an offset automatically. The drive calculates an offset that guarantees that the motor will not have to backup to get to the offset position. This is very convenient for unidirectional applications.

The calculated offset is the distance travelled during deceleration ramp from the home velocity to a stop plus the distance travelled at the home velocity for 1.6msec. This extra distance is used to guarantee that the motor will not need to backup after the deceleration ramp.

The Specified Offset allows the user to choose an exact offset from the Home Reference.

Once the home reference is detected, the device will do whatever is necessary to reach the offset position. This may be as simple as a deceleration to a stop, a continuation at speed followed by a deceleration to a stop, or a deceleration followed by a move in the opposite direction.

To enter a specified home offset, select the Specified Offset radio button. PowerTools FM-3 always displays the calculated offset value as a reference. If the home reference is detected before the axis has reached its peak velocity, the axis will still continue to the precise offset position.

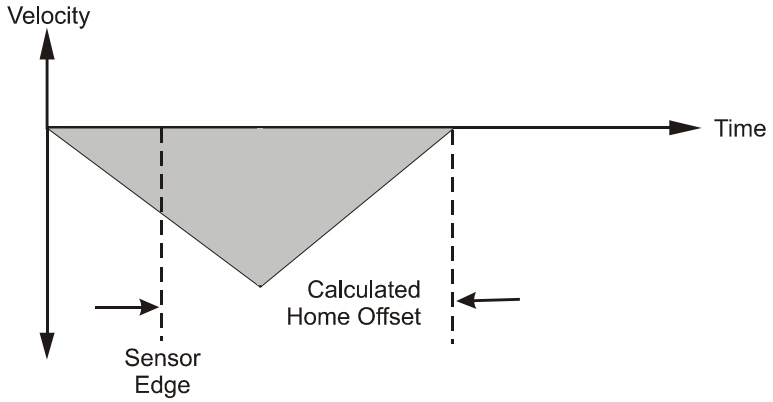


Figure 12: Calculated Home Offset, Peak Velocity Not Reached

If the Home Reference is detected after the axis has reached its peak velocity, the axis will decelerate to the precise offset position.

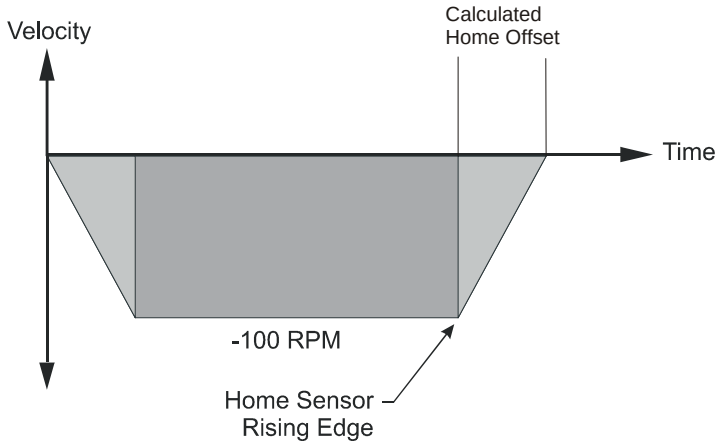


Figure 13: Calculated Home Offset, Peak Velocity Reached

Two examples below show operation when the specified offset is greater or less than the calculated offset. This causes the axis to continue on at speed before decelerating and stopping at the offset position, or backing up after the home sensor.

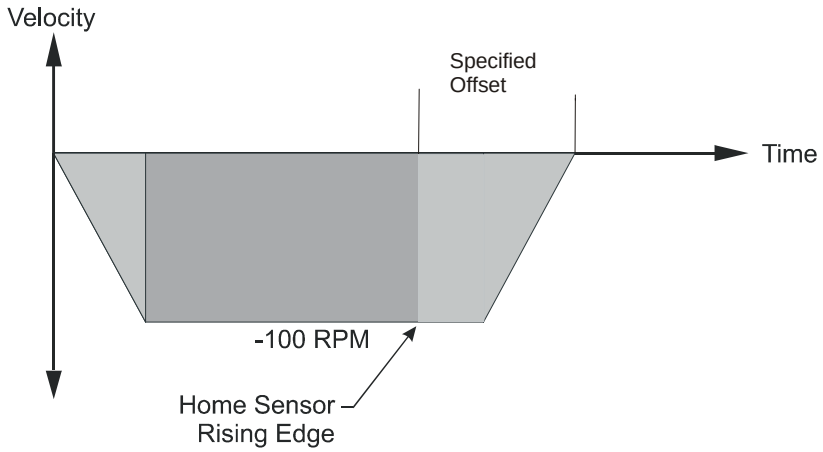


Figure 14: Specified Home Offset, Greater than Calculated Offset

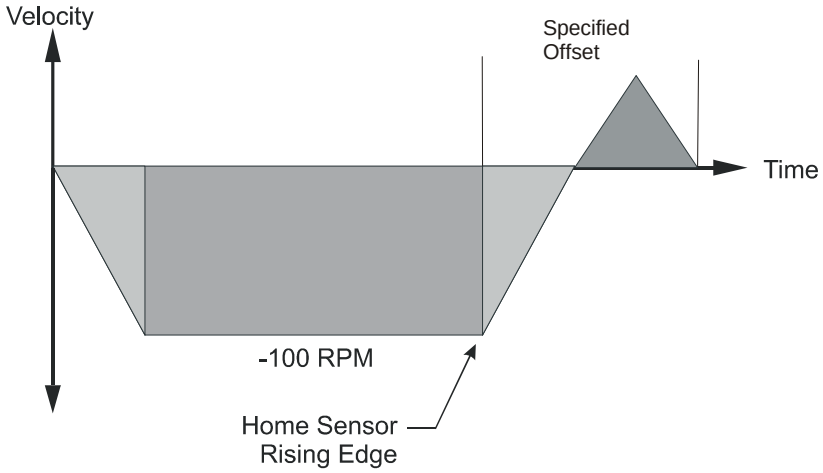


Figure 15: Specified Home Offset, Backup Required

End of Home Position

The End of Home Position (End Posn) defines the home position in relation to the machine's coordinate system. At the completion of the home, the value of the End of Home Position is put into the command position.

Home Limit Distance

This parameter places an upper limit on the incremental distance the motor will travel during the home.

If no reference is found, the system will decelerate and stop at the limit distance. The Home Limit Distance Hit function will be activated if the home stops at the limit distance without finding the reference. Additionally, the Home.CommandComplete function will not turn "On" if the limit distance is hit.

Home Examples

Example 1: Linear Application

In this example, the system uses an external sensor and the motor's encoder marker channel to establish a Home Reference Position. This is the most accurate and most common way to home.

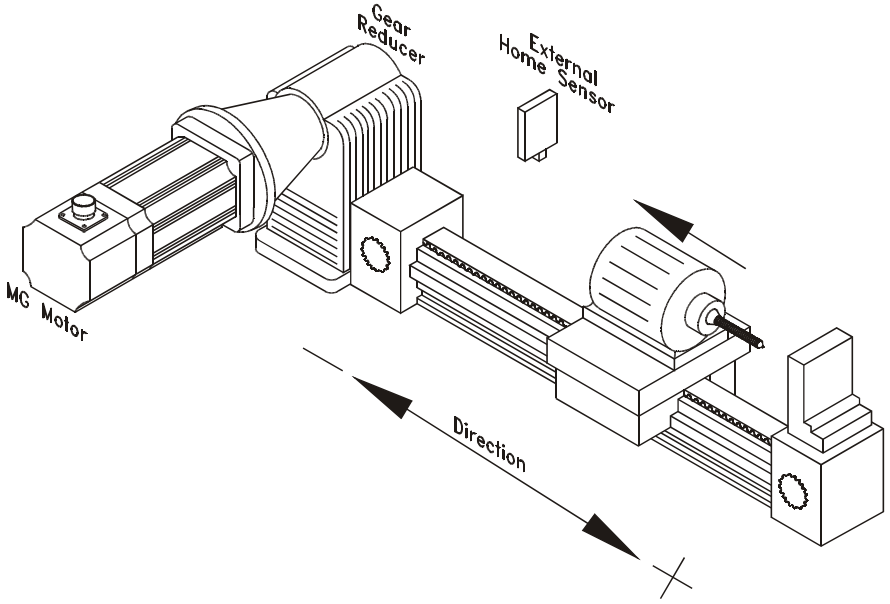


Figure 16: Home to Sensor and Marker, Example

When the FM-3 sees the Home Initiate, it accelerates the motor to the Home Velocity.

The motor continues at that velocity until it first senses the Home Sensor input. It continues at the same velocity until the motor's encoder marker channel is sensed. The rising edge of the motor's encoder marker channel is used to establish the reference position. Once the home reference is detected, the motor decelerates to a stop and moves to the offset position.

Home Sequence

1. If on sensor then back off (if enabled)
2. Search for sensor
3. Search for marker
4. Go to offset (2.0 Revs)
5. Set feedback position to equal End of Home Position

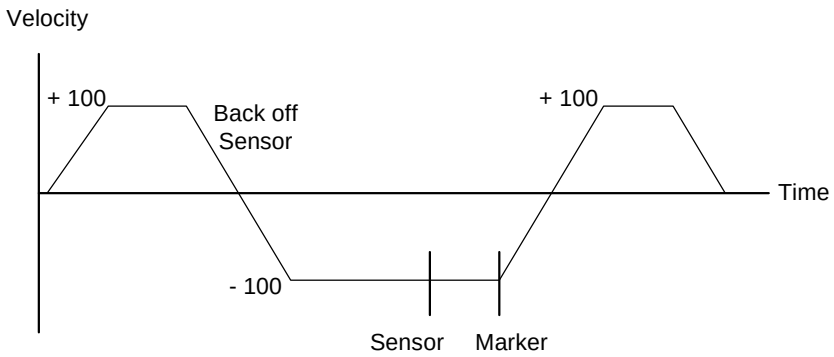


Figure 17: Home Velocity Profile

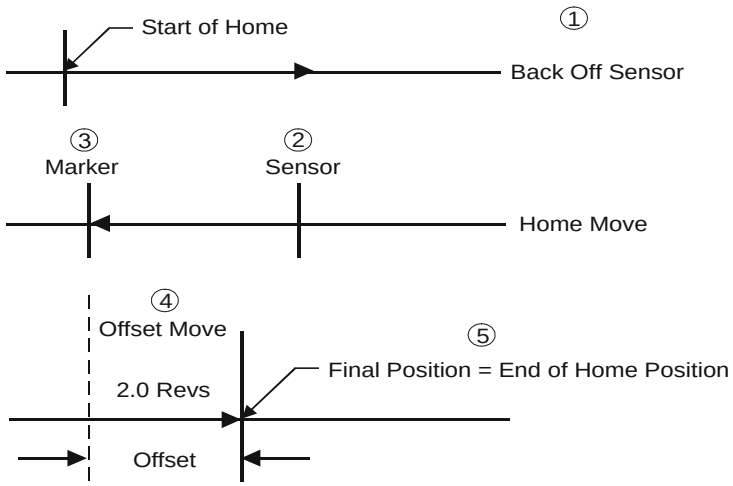


Figure 18: Home Move Sequence

Example 2: Rotary Application

This example uses an external sensor and the motor's encoder marker pulse to establish a home reference position.

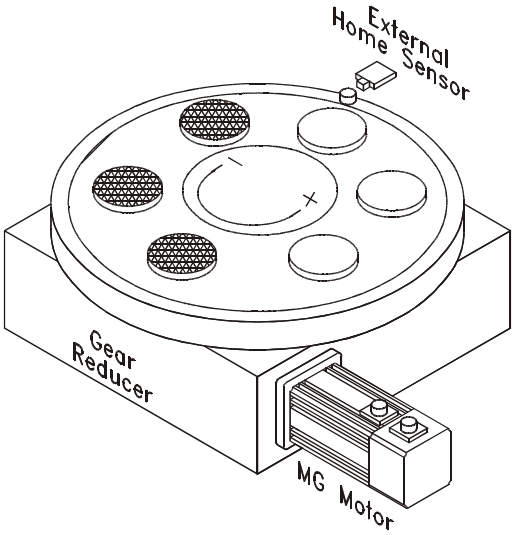


Figure 19: Home Sensor and Marker then Offset, Example

When the device sees the rising edge of the Home Initiate function, it accelerates the motor to the Home Velocity. The motor continues at that velocity until it first senses the Home Sensor input. The motor continues on at the home velocity until the marker is activated.

The rising edge of the motor's encoder marker channel is used to establish the reference position.

After sensing the rising edge of the motor's marker channel, the device will continue moving and will decelerate to a stop at the specified offset position.

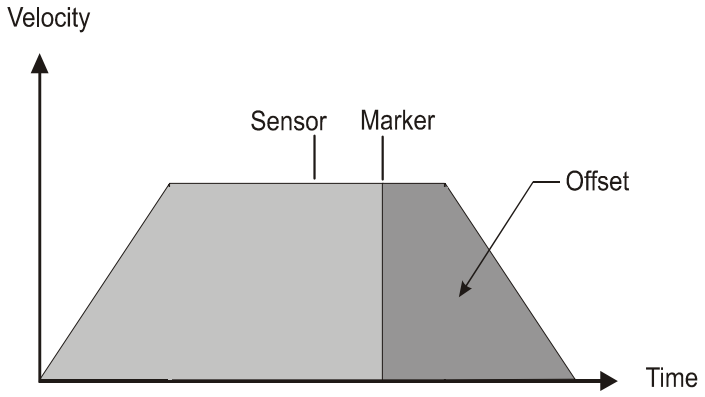


Figure 20: Home Velocity Profile

How Indexes Work

An index is a complete motion sequence that moves the motor a specific incremental distance or to an absolute position. This motion sequence includes an acceleration ramp to a programmed velocity, a run at velocity, and a deceleration to a stop.

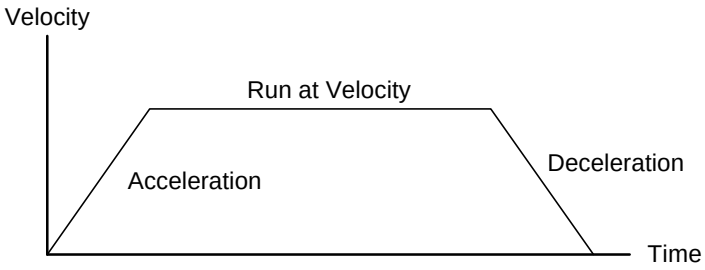


Figure 21: Index Motion Sequence

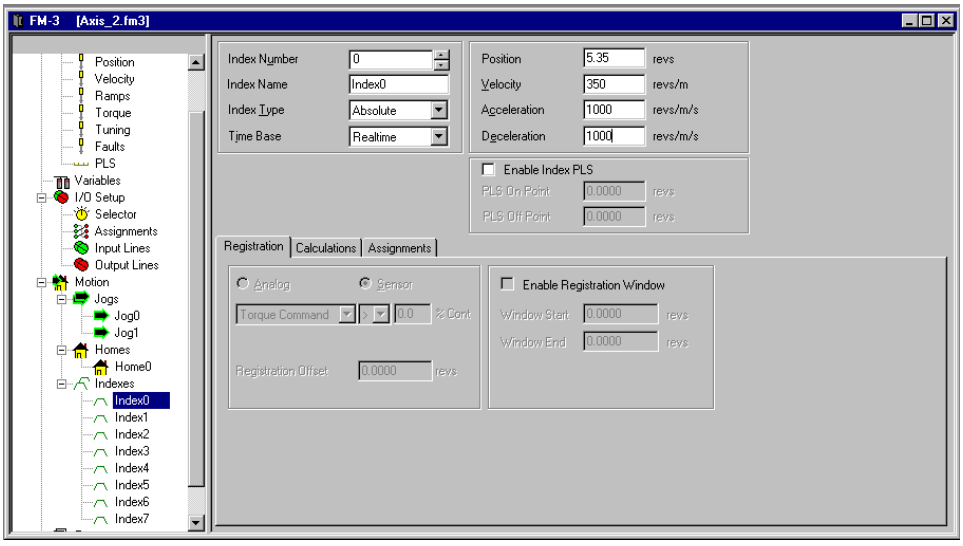


Figure 22: Indexes View

Indexes use acceleration and deceleration ramps which may or may not reach the specified velocity depending on the total distance and the ramp values. For example, a short move with long acceleration and deceleration ramps may not reach the target velocity entered.

Indexes cannot be initiated when any other motion (jogging, homing, or program) is in progress. Indexes can be aborted with the Stop destination found in the Ramps group on the Assignments View.

The FM-3 supports five types of indexes: absolute, incremental, registration, rotary plus and rotary minus.

Absolute vs. Incremental

The difference between absolute and incremental indexes is that absolute indexes move to a specific absolute position and incremental indexes move the motor a specific distance. The figures and explanations below demonstrate this concept.

Note

Home must be executed before an Absolute index will run.

Absolute Indexes

Absolute indexes are used in applications where the motor must travel to a specific position, regardless of where the motor is when the index is initiated.

The FM-3 calculates the distance required to move to the specified position from the current position.

Absolute Index

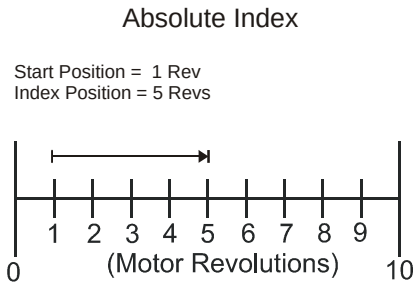


Figure 23: Absolute Index Example 1

In the example above, the current position is 1 rev. If this index is initiated, the motor will travel to a position of 5 revs no matter where it is sitting before the move. From 3 revs, it will travel 2 revs to finish at 5 revs. If the absolute index to 5 revs is initiated a second time immediately after the index, no motion will occur because the motor will already be at a position of 5 revs.

The direction of an Absolute Index is determined by the starting position and the absolute index position. If the starting position for the above index is 9 revs, then the motor will rotate in the negative direction to end up at 5 revs. The diagram below shows this.

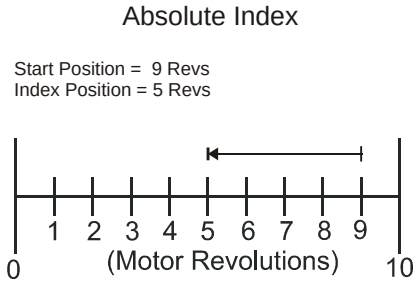


Figure 24: Absolute Index Example 2

Absolute indexes with Rotary Rollover enabled will take the shortest path to the position entered in the index position parameter.

Note

Absolute indexes move to positions relative to where the machine was homed using the Home, or the DefineHome destination.

Incremental Indexes

An incremental index will move the motor a specified distance in the + or - direction regardless of the starting position. The direction of the incremental index motion is determined by the sign (+ or -) of the Index Distance parameter.

Incremental Index

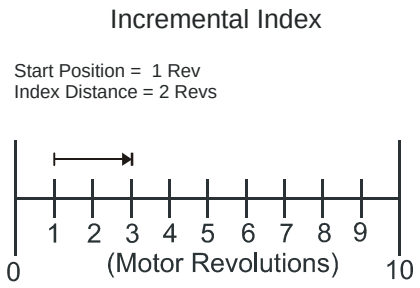


Figure 25: Incremental Index Example

In the example above, the motor starts at 1 rev, travels a distance of 2 revs and stops at 3 revs. If the same index is initiated a second time, the FM-3 would move the motor another 2 revs to a position of 5 revs. If initiated a third time, the motor would travel another 2 revs to a final position of 7 revs. The example below shows this operation.

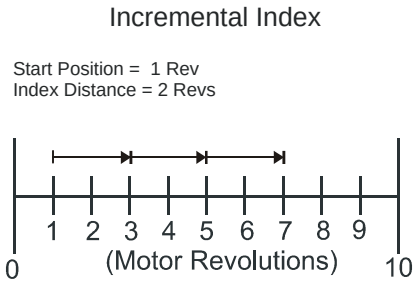


Figure 26: Incremental Index Example 2

Registration Index

A Registration Index is used in applications where the motor must move until an object is detected and then move a specific distance from the point of detection, such as finding a registration mark and moving a distance beyond.

The Registration Index consists of two parts. The first part accelerates the motor to the target velocity and continues at this velocity until it receives a registration trigger (sensor or analog). Upon receipt of a registration trigger, the registration offset will be executed at the target velocity. The Sensor Limit Distance Hit source can be used to turn on an output, if a sensor input or analog limit is not received within the Limit Distance. A registration window can

also be used to determine the validity of a registration trigger. If a registration trigger is received outside of the registration window, it will be ignored.

Rotary Plus and Rotary Minus Indexes

Rotary Plus and Rotary Minus Indexes provide forced directional control of moves to absolute positions. The position entered for a Rotary Plus or Minus type index must be within the rotary range (i.e. 0 ≤ Position < Rotary Rollover Point). All other parameters function the same as they do with absolute indexes. An Absolute Index is a direct move to a specific position, regardless of the starting point. A Rotary Plus Index moves to the specified position, but is forced in a positive direction. Similarly, a Rotary Minus index moves to the specific position, but is forced in a negative direction.

Rotary Plus and Minus Indexes are usually used in rotary applications, therefore the rotary rollover feature on the Setup - Position view in the PowerTools FM-3 software must be enabled to use them.

1. In the following examples the term “D” = (absolute position specified) - (current position). If “D” is negative, motion in the negative direction is implied.

2. In the following examples the Rotary Rollover parameter on the Setup - Position view is set to 360.00°. This means that with each revolution of the motor (or rotary table), feedback will count up to 359.99°, then roll over to Ø°.

Indexes with Rotary Rollover Enabled

Incremental move distances can be outside of the rotary rollover range. See the Setting Up Parameters section for an explanation of Rotary Rollover.

Example 1: If the starting position is at Ø° and 720° is the specified distance, an Incremental index would move 2 revolutions in the positive direction. At the completion of this index the motor position would be Ø°.

Absolute indexes will take the shortest path to the specified position. Absolute index positions must be within the rotary rollover range.

Example 2: If the starting position is at 90° and 80° is the specified position, an Absolute index would travel 10° in the negative direction. At the completion of this index the motor position would be 80°.

Example 3: If the starting postion is 45° and 315° is the specified postion, an Absolute index would travel 90° in the negative direction because that is the shortest path between 45° and 315°.

Rotary Plus indexes will move to the specified position and are forced in a positive (or plus) direction. Rotary Plus index distances must be within the rotary rollover range.

Example 4: As in example 2 above, the starting position is at 90° and 80° is the specified position. A Rotary Plus index would travel 350° in the positive direction. At the completion of this index the motor position would be 80°.

Example 5: If the starting postion is 10° and the specified postion is 350°, a Rotary Plus index will travel 340° in the positive direction.

Rotary Minus indexes move to the specified position, but are forced to travel in the negative (or minus) direction. Rotary Minus index postions must be within the rotary rollover range.

Example 6: As in examples 2 and 4 above, the starting position is at 90° and 80° is the specified position. A Rotary Minus index would travel 10° in the negative direction. At the completion of this index the motor position would be 80°.

Example 7: If the starting postion is 15° and the specified postion is 270°, a Rotary Minus index would travel 105° in the negative direction.

How Communications Work

Downloading

Downloading is the process of sending your configuration created with PowerTools FM-3 from the PC to the FM-3 module. Changes made in PowerTools FM-3 will not take effect until the information has been downloaded or the Update to RAM button has been clicked.

To download information to a FM-3, click on the download button on the PowerTools toolbar or select download from the Device pull down menu. PowerTools will lead the user through a series of dialog boxes that determine what baudrate will be used and which communications port on the user's PC will be used.

Uploading

Uploading is the process of reading information back from the drive to the PowerTools FM-3 configuration screens.

To upload information from a drive, click on the upload button on the PowerTools toolbar or select upload from the Device pull down menu. This leads the user through a series of dialog boxes that determine which drive address is to be uploaded.

Updating to RAM

The Update to RAM button can be used to send changes to the FM-3 without performing a complete download. The Update to RAM button is found in the PowerTools FM-3 toolbar. This operation will send only those changes that have been made since the last Update to RAM or Device – Download to the FM-3. The changes will take effect immediately upon clicking on the button.

Caution

The parameters will be sent to the FM-3 without stopping motion or disabling the drives. Because of this, it is important to use caution when changing motion parameters while the motor is in motion.

The Update to RAM button saves the parameters only to RAM and not to Non-Volatile Memory (NVM). Therefore, if the system power is removed, any changes made using the Update to RAM button will be lost. In order to save changes to NVM, a full-download must be performed.

The flowchart below describes a typical process using the Update to RAM to make changes, and then downloading when complete to save changes to NVM.

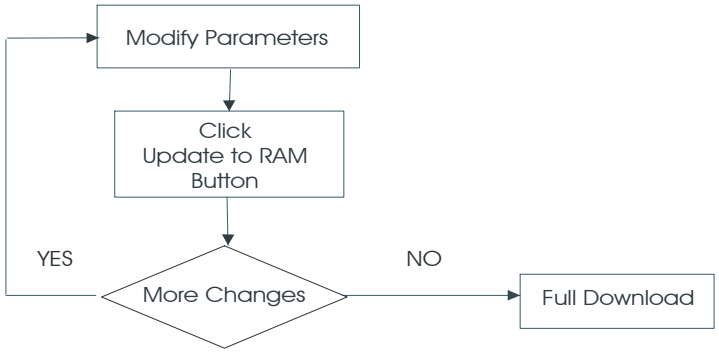


Figure 27: Update to RAM Flow Chart

The Update to RAM button operates according to the following rules:

If no parameters have been modified by the user, the Update to RAM button will be disabled.

If the user modifies a parameter that does not require a full download, the Update to RAM icon will be enabled.

If while the icon is enabled, the user modifies a parameter that requires a full download, the Update to RAM icon will become disabled.

When the user clicks on the Update to RAM icon, all the modified parameters are transmitted to the FM-3. Once transmitted, the icon will become disabled again until another parameter is changed.

If the user performs a full download while the button is enabled, when the download is complete, the Update to RAM button will be disabled.

If the user modifies parameters, and disconnects, the update button will be disabled, and the changes will not be sent.

Setting Up Parameters

Setup View

The Setup View contains all of the primary system setup parameters. These parameters must be setup prior to using your system.

By selecting Setup in the hierarchy view, the Setup view will appear in the right side of the view (see Figure 28). The right part of the view is divided into groups, with an explanation of each function. The groups are Identification, Configuration, Drive Encoder Output, Positive Direction, and Status Online.

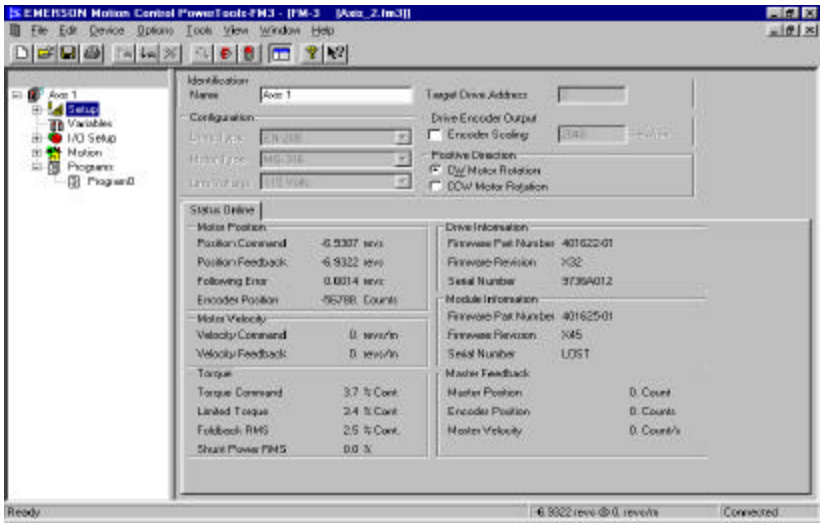


Figure 28: Setup View

Identification Group

The identification group consists of the Device Name and the Target Drive Address.

Name

This is a 12-character alpha/numeric user-configured name for this axis. Enter this name for the device you are currently setting up. Assigning a unique name for each device in your system allows you to quickly identify a device when downloading, editing, and troubleshooting. All keyboard characters are valid. This will default to Axis 1.

Target Drive Address

This is the Modbus address of the target drive to which you will download the configuration. The default target drive address is 1.

Configuration Group

The configuration group consists of list boxes for Drive Type, Motor Type, and Line Voltage.

Drive Type List Box

Select the drive model for the system you are currently setting up.

Motor Type List Box

Select the motor you want to use. Emerson Motion Control PowerTools® FM-3 software will only display the motor models that are compatible with the drive you selected and any user defined motors.

Caution

Selecting the wrong motor type can cause poor performance and may even damage the motor and/or drive.

Line Voltage List Box

Line Voltage specifies the applied power and adjusts the internal gains to compensate for it. This parameter has two choices: 115 VAC and 230 VAC. If the Line Voltage is set to 230 VAC when the actual applied voltage is 115 VAC, the motor will be slightly less responsive to commands and load disturbances.

Caution

The Line voltage must never be set to 115 VAC if the applied voltage is actually 230 VAC. This can cause drive instability and failure.

Drive Encoder Output Group

The drive encoder output group consists of the encoder scaling check box and encoder scaling.

Encoder Scaling Check Box

This check box enables the Scaling parameter of the Drive Encoder Output.

Encoder Scaling

This parameter defines the encoder resolution (lines per revolution) of the drive's encoder output. This feature allows you to change the drive encoder output resolution in increments of 1 line per revolution up to the density of the encoder in the motor. If the Encoder Output

Scaling parameter is set to a value higher than the motor encoder density, the drive encoder output density will equal that of the motor encoder.

Positive Direction Group

The Positive Direction group consists of a clockwise (CW) Motor Rotation Radio Button or a counter-clockwise (CCW) Motor Rotation Radio Button.

The motion will move in either CW direction or counter-clockwise CCW direction. Perspective of rotation is defined as you face the motor shaft from the front of the motor.

CW Motor Rotation Radio Button

Select this radio button for applications in which CW motor rotation is considered to be motion in the positive direction (increasing absolute position).

CCW Motor Rotation Radio Button

Select this radio button for applications in which CCW motor rotation is considered to be motion in the positive direction (increasing absolute position).

Status Online Tab (Online Only)

The Status Online tab is visible when online and consists of the Motor Position group, Drive Information group, Motor Velocity group, Module Information group, Master Feedback group, and the Torque group (shown in figure 26).

Motor Position Group

Position Command

Position command is the commanded motor position sent to the EN drive by the FM-3. This parameter does not take following error into account. See also PosnFeedback and FollowingError. Units are in user units.

Position Feedback

Feedback position is the actual motor position in user units. PosnCommand minus the PosnFeedback is the FollowingError

Following Error

The Following Error is the difference between the Position Command and the Position Feedback. It is positive when the Position Command is greater than the Position Feedback.

Encoder Position

Motor encoder position in encoder counts. This position reflects the feedback position of the motor and is not scaled into user units. This is a signed 32 bit value.

Drive Information Group

Firmware Part Number

Displays the part number of the drive firmware you are using.

Firmware Revision

Displays the revision of the drive firmware you are using.

Serial Number

Displays the serial number of the drive with which you are currently online.

Motor Velocity Group

Velocity Command

The Velocity Command is the velocity that the FM-3 drive is commanding the motor to run at. This command is generated by the drive velocity control loop. It is displayed in user units.

Velocity Feedback

This is the feedback (or actual) velocity. It is calculated using the change in position of the motor encoder. It will always return the actual motor velocity - even in synchronized applications in which the master axis is halted during a move.

Module Information Group

Firmware Part Number

Displays the part number of the FM-3 firmware you are using.

Firmware Revision

Displays the revision of the firmware in the FM-3 with which you are currently online.

Serial Number

Displays the serial number of the FM-3 with which you are currently online.

Master Feedback Group

Master Position

Used for synchronized motion, this displays the position of the master encoder in units defined on the Master Units Setup View.

Encoder Position

This displays the position of the master encoder in counts.

Master Velocity

This displays the velocity of the master encoder in master units/second.

Torque Group

Torque Command

This displays the torque command value before it is limited. The torque command may be limited by either the Torque Limit (if the Torque Limit Enable destination is active) or current foldback. Units for this parameter are defined in the Torque Group on the User Units View.

Limited Torque

This is the actual torque commanded to the motor. This value is the result after the TorqueCommand is limited by the current foldback or the TorqueLimit value (if enabled).

Foldback RMS

This parameter accurately models the thermal heating and cooling of the drive and motor. When it reaches 100 percent, current foldback will be activated. See the Diagnostics section for an explanation of foldback.

Shunt Power RMS

This parameter models the thermal heating and cooling of the drive internal shunt. This parameter indicates the percent of shunt capacity utilization. When this value reaches 100 percent, the drive will generate an RMS Shunt Power Fault. This parameter is not applicable to the EN-204 which does not have an internal shunt resistor. This parameter is applicable to the EN-208 and EN-214.

User Units View

The User Units View is used to scale the desired application units into known values. All information for distance, velocity, and accel/decel units are set up here and used throughout the system setup.

By selecting User Units in the hierarchy view, the User Units View will appear on the right (see Figure 29).

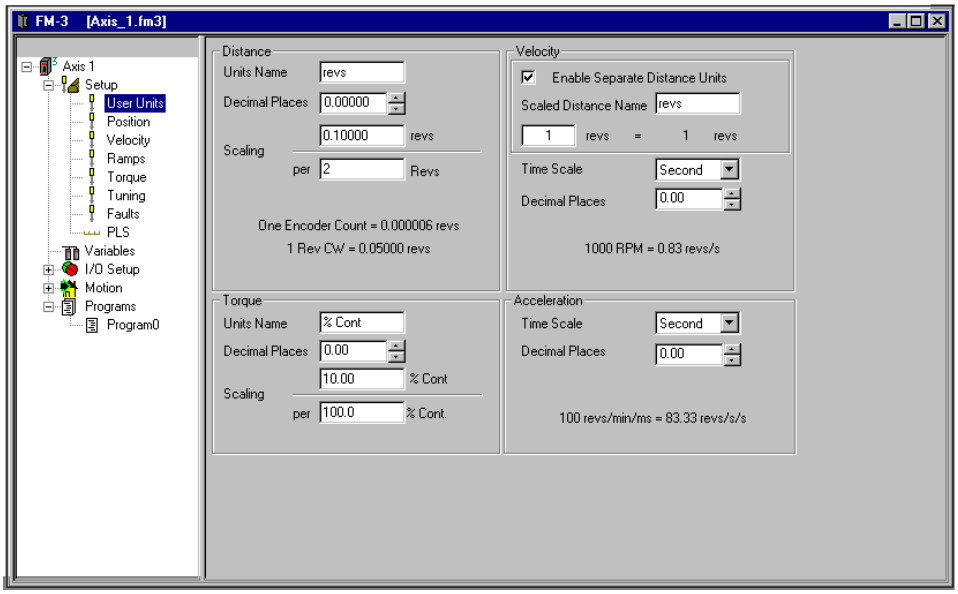


Figure 29: User Units View

Distance Group

Units Name

This is a 10-character name for the distance user units you want to use in your application.

Decimal Places

The number of decimal places set in this parameter determines the number of digits after the decimal point used in all distance and position parameters throughout the software. Using a high number of decimal places will improve position resolution, but will also limit the maximum position. You can select from zero to six decimal places of accuracy.

Scaling

A Characteristic Distance and Length must be established to allow the FM-3 to scale user units back to actual motor revolutions. This scaling factor is as follows:

$$\text{Scaling} \Rightarrow \frac{\text{Characteristic Distance}}{\text{Characteristic Length}}$$

Characteristic Distance

This is the distance the load travels (in user units) when the motor travels the characteristic length (in motor revolutions).

Characteristic Length

This is the distance the motor travels (in whole number of revolutions) to achieve one characteristic distance of load travel.

Distance Scaling Examples

A 1.5" diameter pulley is used to drive a conveyor belt, and the user wishes to use units of inches instead of revolutions.

Units Name —> Set to Inches

Decimal Places —> Set to desired accuracy 0.000

In one revolution of the motor (or pulley), the belt will travel a distance of one pulley circumference.

$= 1.5" \times \pi$

$= 1.5 \times 3.14...$

$= 4.712$ inches / revolution

$$\Rightarrow \frac{\text{Characteristic Distance} = 4.712}{\text{Characteristic Length} = 1}$$

If the user decides to put a 5:1 reducer on the system, they simply need to change the characteristic length.

Now the belt travels 4.71" in 5 motor revolutions.

$$\Rightarrow \frac{\text{Characteristic Distance} = 4.712}{\text{Characteristic Length} = 5}$$

Keep in mind that the characteristic length is always a whole number and the valid range is from 1 to 2000.

Velocity Group

Enable Separate Distance Units Check Box

If checked, separate distance and velocity units, name and scaling will be enabled. If not enabled, the velocity units, name and scaling will be defined by the Distance Group.

Scaled Distance Name

If the user wants the velocity units to have a different distance scaling than the distance units a name can be entered here up to 10 characters. For example, the user distance units could be inches while the velocity units are feet per minute.

Velocity Distance Units Scale Factor

This parameter scales the Velocity Distance Units back to actual distance units. To do this, enter the number of distance user units that are equal to one velocity scaled distance unit.

Separate Distance Units Example:

A user has an application using a leadscrew with a 0.5"/turn lead. The user wants to have Distance Units of Inches, but wants Velocity Units of Feet so motion can be programed in feet/minute.

Distance Units Name —> Set to Inches

Enable Separate Distance Units —> Enabled (checked)

Scaled Distance Name —> Set to Feet

Velocity Distance Units Scale Factor —> # of Distance Units / 1 Scaled Distance Unit

1 Foot = 12 Inches

=> Velocity Distance Units Scale Factor = 12

Time Scale List Box

The time can be one of two values: seconds or minutes. This selection sets the real-time velocity time scale.

Decimal Places

The number of decimal places set in this parameter determines the number of digits after the decimal point used in all real-time velocity parameters throughout the software. Using a high number of decimal places will improve velocity resolution, but will also limit the maximum velocity. You can select from zero to six decimal places of accuracy.

Acceleration Group

Time Scale List Box

From this list box, select the acceleration timescale to be used for all real-time profiles. The time scale selected will be used for both acceleration and deceleration parameters. You can select from milliseconds or seconds.

Decimal Places

The number of decimal places set in this parameter determines the number of digits after the decimal point used in all real-time accel/decel parameters throughout the software. Using a high number of decimal places will improve accel/decel resolution, but will also limit the maximum accel/decel rate. You can select from zero to six decimal places of accuracy.

Torque Group

Units Name

10-character name for the torque user units.

Decimal Places

The number of decimal places set in this parameter determines the number of digits after the decimal point used in all torque parameters throughout the software. Using a high number of decimal places will improve torque resolution, but will also limit the maximum torque. You can select from zero to six decimal places of accuracy.

Scaling

The amount of torque in user torque units will be set equal to the Percent Continuous Current. This parameter is used to scale the actual torque back into the user defined units. The units of this parameter are % ContinuousCurrent. This scaling factor is used along with the user torque to establish a relationship between torque user units and actual torque.

Master Units View

Master Units setup provides the setup parameters for use with synchronized motion. This setup window determines how the encoder signals are interpreted and establishes the scaling for all master units (master distance, master velocity, etc.).

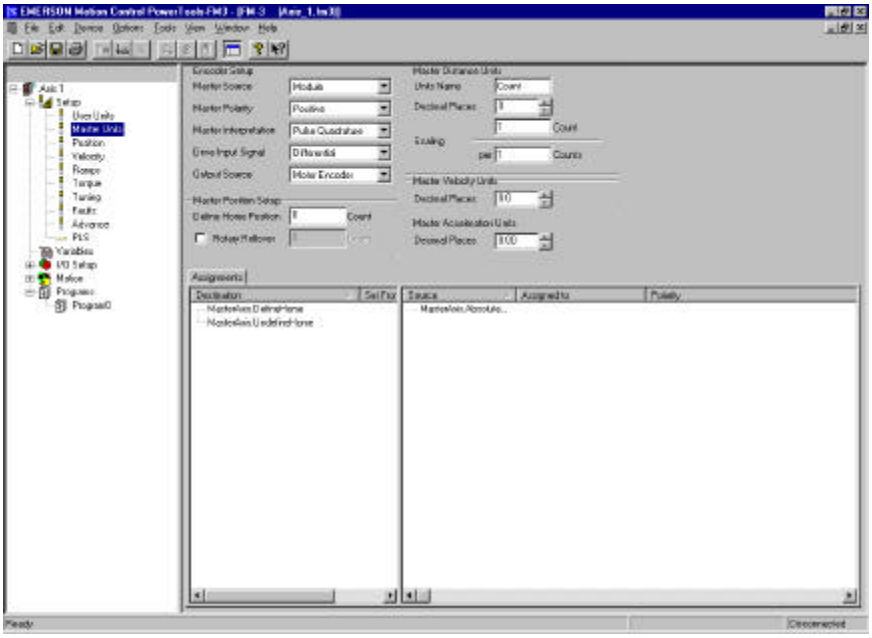


Figure 30: Master Units View

Encoder Setup Group

Master Source

Master Source indicates the hardware location of the master encoder input. Select module to use the sync input connector on the FM-3; select drive to use the drive 44-pin command connector on the E Series drive.

Master Polarity

Master Polarity defines the direction of the master encoder that corresponds to a positive master position change.

Master Interpretation

Master Interpretation determines how the incoming pulses are seen to generate the synchronized motion command. This setting allows you to choose the appropriate signal type to match the device generating the master input pulses.

Drive Input Signal

Drive Input Signal is selected accordingly based on whether the incoming pulses are Differential (default) or Single Ended.

Output Source

Output Source determines which signal will be sent to the Sync Output connector on the FM-3. If Motor Encoder (default) is selected, then the encoder signals from the motor that the FM-3/En is controlling will be sent out the FM-3 Sync Output connector. If Drive Encoder Input is selected, then the synchronization signals sent to the Drive 44-pin command connector will be sent to the FM-3 Sync Output connector. If Module Encoder Input is selected, then the same signal coming into the FM-3 Sync Input connector will be sent out the Sync Output connector.

Master Position Setup Group

Define Home Position

Define Home Position is the value that the Master Position Feedback will be set to when the MasterAxis.DefineHome destination is activated. After the MasterAxis.DefineHome has been activated, the MasterAxis.AbsolutePosnValid source will activate.

Rotary Rollover Check Box

If checked, the rotary rollover feature for the Master Axis will be enabled.

Rotary Rollover

If enabled, the Master Position will rollover to zero at the value specified here. As the master encoder counts up, the master position feedback will increase until it reaches the rollover value and then reset to zero and continue to count up. If rotating in the negative direction, the master position feedback will decrease until it reaches zero, and then start over at the Rotary Rollover value.

Master Distance Units

The parameters in this group are used to establish the scaling of the master axis into user units.

Units Name

This is a text string up to 12 characters that will be used to define the units of distance traveled by the master axis for incoming synchronization signals.

Decimal Places

The number of decimal places set in this parameter determines the number of digits after the decimal point used in all distance and position parameters used in synchronized motion throughout the software. Using a high number of decimal places will improve position resolution, but will also limit the maximum position. You can select from zero to six decimal places of accuracy.

Scaling

The scaling factor is defined as MasterAxis.CharacteristicDistance / MasterAxis.Counts. The numerator (top value of the scaling fraction) is the Characteristic Distance. The denominator (bottom value of the scaling fraction) is the # of Counts. The Characteristic Distance is the number of Master Distance Units that will be traveled per number of counts in the bottom of the fraction. The Counts parameter is the number of incoming pulses it takes to travel the characteristic distance.

Master Velocity Units

Decimal Places determines the number of decimal places to be used in the velocity parameter for all synchronized motion.

Velocity Filter

The FM-3 uses a Low Pass Filter to smooth the observed velocity of the master encoder signal. For synchronized motion, this filter helps to reduce noise in the command changes to the drive which are based on the observed change of the master encoder. The user can set the bandwidth (in Hz) of this filter based on the anticipated dynamics of the master encoder. For a highly dynamic master, the bandwidth should be increased. For stable, slowly changing masters, the bandwidth can be lowered.

Master Acceleration Units

The number of decimal places set in this parameter determines the number of digits after the decimal point used in all real-time accel/decel parameters used for synchronized motion throughout the software. Using a high number of decimal places will improve accel/decel resolution, but will also limit the maximum accel/decel rate. You can select from zero to six decimal places of accuracy.

Position View

The Position View allows you to set up and view the parameters related to drive positioning.

See Figure 31. The left part of the view shows that Position has been selected. The right part of the view is divided into groups. An explanation of the groups and their functions is provided below.

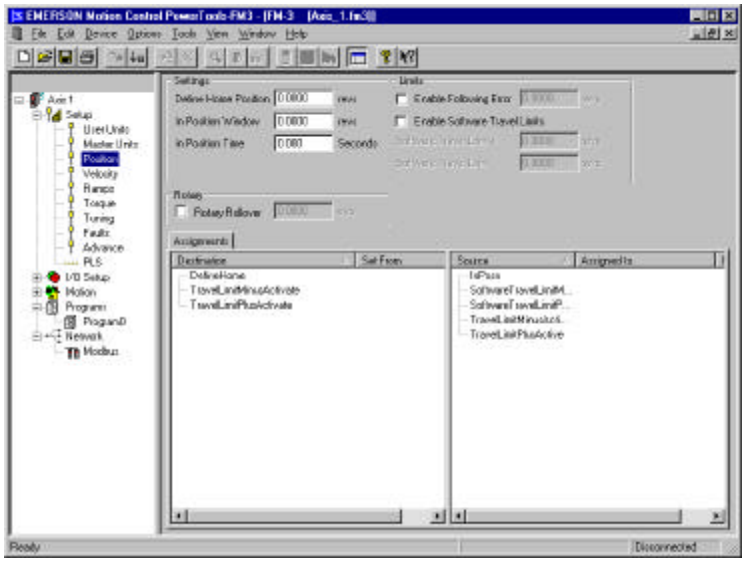


Figure 31: Position View

Settings Group

Define Home Position

This is the value to which the position command will be set when the Define Home destination is activated. This is used in applications which do not use a home, but is required to define a known reference point. The units are defined on the User Units View.

In Position

The In Position source will activate at the end of a move if the absolute value of a following error is less than or equal to the In Position Window for the In Position Time.

In Position Window

The absolute value of the Following Error must be less than or equal to this value at the end of an index in order for the InPosn destination to activate. This window is set in units specified in the User Units View.

For example:

The In Position window is set to .0025 revs. At the end of an index, the following error is calculated to be .0012 revolutions. Therefore, the InPosn source will activate.

In Position window is set to .001 inches. If at the end of an index, the following error is calculated to be .0015 inches, then the InPosn source will not activate.

In Position Time

This is the amount of time in seconds that commanded motion must be complete and the absolute value of the following error must be less than the In Position Window for the InPosn source to activate. If set to zero (default), then InPosn will activate as soon as motion stops and the following error is less than the In Position Window parameter.

Limits Group

Enable Following Error Check Box

Check this box to enable (or disable if not checked) the Following Error Limit. If enabled, a fault will be generated if the absolute value of the following error ever exceeds the value in the following error parameter. If disabled, a fault will never be generated.

Following Error

The Following Error is the difference between the Position Command and the Position Feedback. It is positive when the Position Command is greater than the Position Feedback. If the absolute value of the following error exceeds the value you enter here, the drive will generate a Following Error Fault (F). All accumulated Following Error will be cleared when the drive is disabled.

The Following Error Limit is defined in user units.

Enable Software Travel Limits Check Box

Check this box to enable (or disable if not checked) the software travel limits. If unchecked, the software travel limits are not monitored.

Software Travel Limits

Software Travel limits can be used to limit machine travel. They are often setup inside the hardware travel limits to add another level of security or protection from exceeding the machines travel limits. The FM-3 constantly monitors the feedback position, and if this position exceeds the values entered for Software Travel Limit + or -, then the drive will decel to a stop. Software Travel Limits are not functional unless the Absolute PosnValid source is

active. Absolute Position Valid is active upon successful completion of a home or the DefineHome destination is activated.

To recover from a software travel limit, a jog may be commanded in the opposite direction of travel. For example, if a software travel limit - is hit, then the axis can be jogged in the + direction.

Software Travel Limit +

If the absolute position is greater than or equal to this value the Software Travel Limit Plus Active source shall activate.

A rising edge occurs when the absolute position is greater than or equal to the parameter Software Travel Limit +. A falling edge will be generated as soon as the above is not true.

Software Travel Limit -

If the absolute position is less than or equal to this value the Software Travel Limit Minus Activate shall activate.

A rising edge occurs when the absolute position is less than or equal to the parameter Software Travel Limit -. A falling edge will be generated as soon as the above is not true.

Rotary Group

Rotary Rollover Check Box

Check this box to enable (or disable if not checked) the rotary rollover feature.

Rotary Rollover

This parameter is used in rotary applications and determines the position at which the internal position counter will be reset to zero.

Example:

The user has a rotary table application with distance user units of degrees, 360.00 degrees/1 rev. The Rotary Rollover would be set to a value of 360°.

The motor is traveling in the positive direction. As the feedback position reaches 359.999 and continues on, the feedback position will reset (or roll-over) to zero. If the motor changes direction and travels in the negative direction, the position will rollover at 0 to 359.999 degrees and count down. The resolution of the rotary rollover point is determined by the Distance Units Decimal Places parameter on the Setup - User Units view in the PowerTools FM-3 software.

If an absolute index is used with a non-zero rotary rollover point, the FM-3 will calculate the shortest path to its destination and move in the required direction.

To force the motor to run a certain direction, use the Rotary Plus or Rotary Minus type of indexes.

Assignments Tab

Displays any assignments to functions on this view that have been made using the assignments view.

Online Tab

The Online Tab is shown in Figure 31. While online, the following real-time data will be displayed.

Motor Position Group

Position Command

This is the commanded position in user units generated by the FM-3.

Position Feedback

This is the feedback position of the motor in user units.

Following Error

The Following Error is the difference (in user units) between the Position Command and the Position Feedback. It is positive when the Position Command is greater than the Position Feedback.

Encoder Position

The motor position in encoder counts since power up when the value was set to zero. This is a signed 32-bit value.

Velocity View

The Velocity View allows the setup of feedrate override details.

By selecting Velocity in the hierarchy view, the Velocity View will appear in the right (see Figure 32).

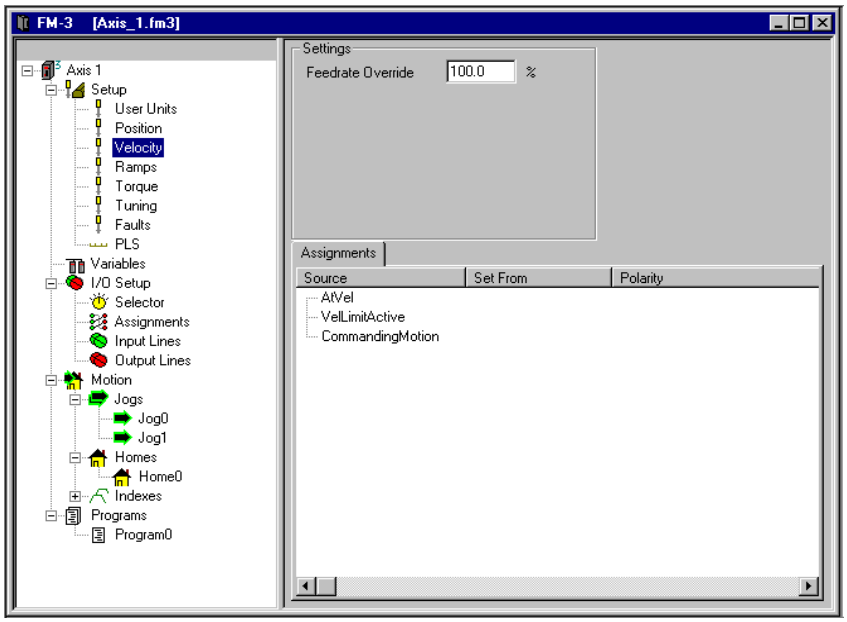


Figure 32: Velocity View (Assignments Tab shown)

Settings Group

Feedrate Override

This parameter is used to scale all motion. It can be described as scaling in real time. The default setting of 100% will allow all motion to occur in real time. A setting of 50% will scale time so that all moves run half as fast as they do at 100%. A setting of 200% will scale time so that all moves run twice as fast as they would at 100%. Feed Rate Override is always active and affects all motion, including accels, decels, and synchronized motion. This parameter may be modified via Modbus or in a program.

Caution

Using the Update to RAM to change the Feedrate Override will implement the changes instantly. Instantaneous acceleration changes will result.

Online Tab (not shown)

If online, this view will show the limits group and the velocity group.

Velocity Group

Velocity Command

The Velocity Command is the actual command generated by the FM-3 velocity loop in user units.

Velocity Feedback

This parameter is the actual feedback motor velocity in user units.

Assignments Tab (shown)

Displays any assignments to functions on this view that have been made on the assignments view.

Ramps View

The Ramps View contains all setup information for the global acceleration and deceleration profiles.

By selecting Ramps in the hierarchy view, the Ramps View will appear in the right (see Figure 33).

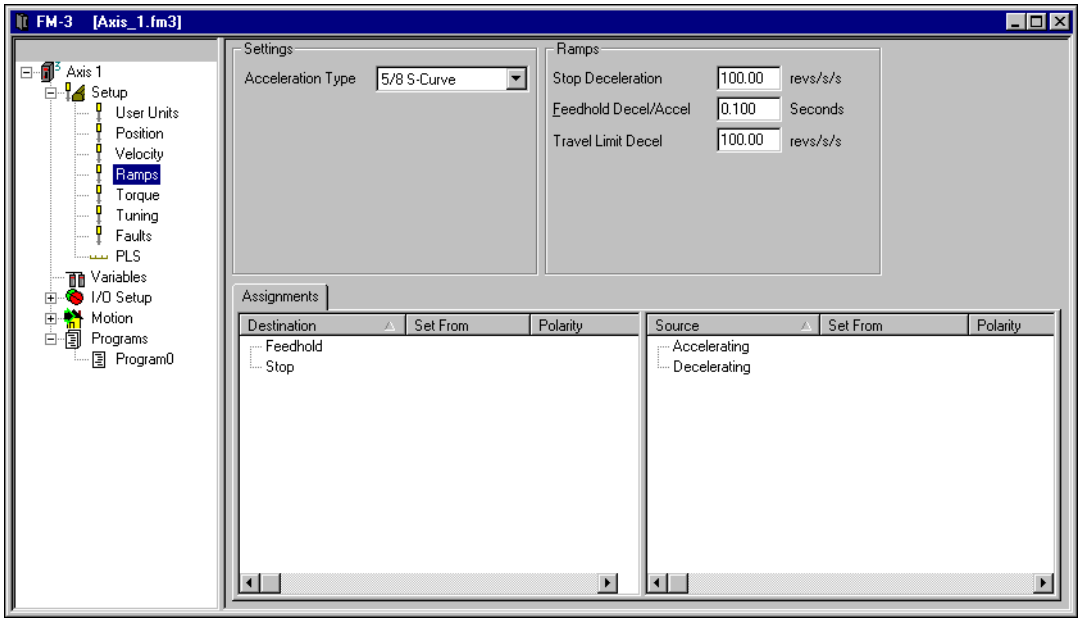


Figure 33: Ramps View

Settings Group

The Settings Group contains the Acceleration Type List Box.

Acceleration Type

Press the arrow by the Acceleration Type list box. It will display the various acceleration types: 5/8 S-Curve, 1/4 S-Curve, Linear, and S-Curve.

This is used to select the acceleration/deceleration type for all motion (homes, jogs and indexes). The “S-Curve” ramps offer the smoothest motion, but lead to higher peak acceleration/deceleration rates. “Linear” ramps have the lowest peak acceleration/deceleration rates but they are the least smooth ramp type. “5/8 S-Curve” ramps and “1/4 S-Curve”

ramps use smoothing at the beginning and end of the ramp but have constant (linear) acceleration rates in the middle of their profiles. The “5/8 S-Curve” is less smooth than the “S-Curve” but smoother than the “1/4 S-Curve”.

S-Curve accelerations are very useful on machines where product slip is a problem. They are also useful when smooth machine operation is critical. Linear ramps are useful in applications where low peak torque is critical. Below is a comparison of the 4 ramp types:

S-Curve: Peak Acceleration = 2 x Average Acceleration

5/8 S-Curve: Peak Acceleration = 1.4545 x Average

1/4 S-Curve: Peak Acceleration = 1.142857 x Average Acceleration

Linear: Peak Acceleration = Average Acceleration

Ramps Group

Stop Deceleration

The value you enter here defines the deceleration rate which is used when the Stop destination is activated. The default is 100 RPM/second.

The Stop destination is found in the Ramps Group in the Assignments view.

Feedhold Decel/Accel

When the Feedhold destination is activated, the motor will decelerate to a stop in the time specified by the FeedholdDecelTime parameter. When feedhold is cleared, the motor will accelerate back to speed in the same specified period of time.

Feedhold is a means to halt the motor within a velocity profile and then return to the profile later at the exact same place in the profile. Feedhold does not ramp and does not decelerate in terms of velocity. Instead, it stops by decelerating time. For example, if the motor is running at 50 revs/second and feedhold is activated with 2 seconds specified in the FeedholdDecelTime parameter, then the motor will actually slow and stop in 2 seconds as measured time (on a time/velocity profile) goes from 100% to 0%.

Travel Limit Decel

The value entered here is the deceleration ramp that is used when a software or hardware travel limit is hit.

Assignments Tab

Displays any assignments to functions on this view that have been made on the Assignments View.

Torque View

The Torque View allows you to edit torque level and limit parameters as well as view realtime torque values when online.

By selecting Torque in the hierarchy view, the Torque View will appear in the right (see Figure 34). The right part of the screen is divided into groups. An explanation of the groups and their functions is provided below.

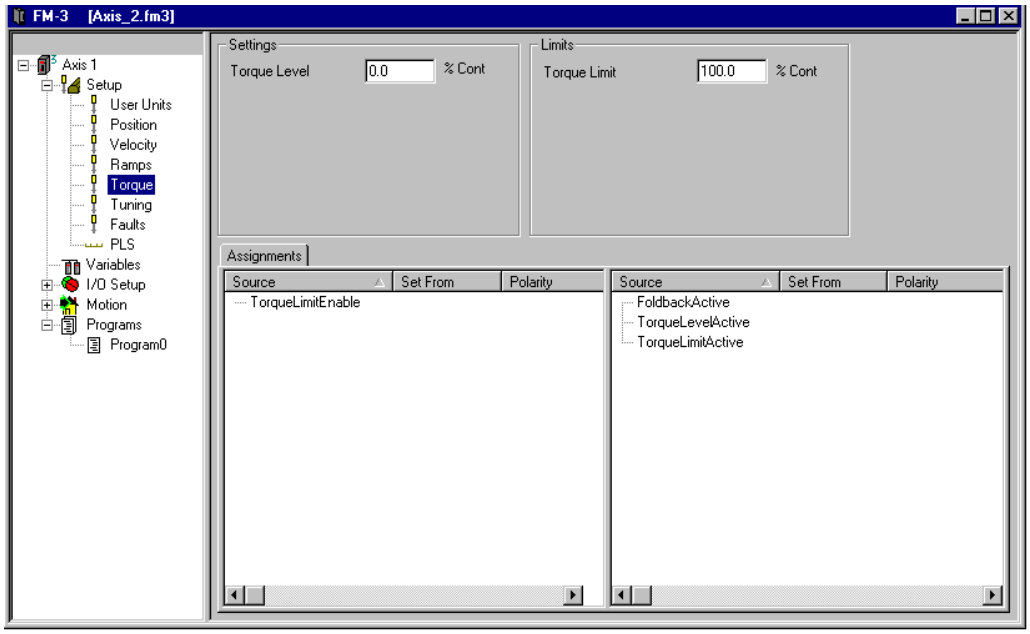


Figure 34: Torque View

Settings Group

Torque Level

This parameter sets the activation point for the Torque Level Active source. If set to 100%, the Torque Level Active source will activate any time the Torque Command reaches or exceeds 100% continuous. This parameter is specified in Torque User Units.

Limits Group

Torque Limit

This parameter sets the value to which the Torque Command will be limited when the Torque Limit Enable destination is active. To make the Torque Limit always active, assign the Torque Limit Enable destination to the Initially Active source on the Assignments view.

Assignments Tab

Displays any assignments to functions on this view that have been made on the Assignments View.

Online Status Tab

If online, this view will show the Torque Command, Limited Torque, Foldback RMS, and Shunt Power RMS.

Tuning View

The Tuning View allows you to modify tuning parameters based on specific application information.

By selecting Tuning in the hierarchy view, the Tuning View will appear in the right (see Figure 35). The right part of the screen is divided into groups. An explanation of the groups and their functions is provided below.

For help on calculating tuning parameters and more in-depth tuning information, turn to “Tuning Procedures” on page 157.

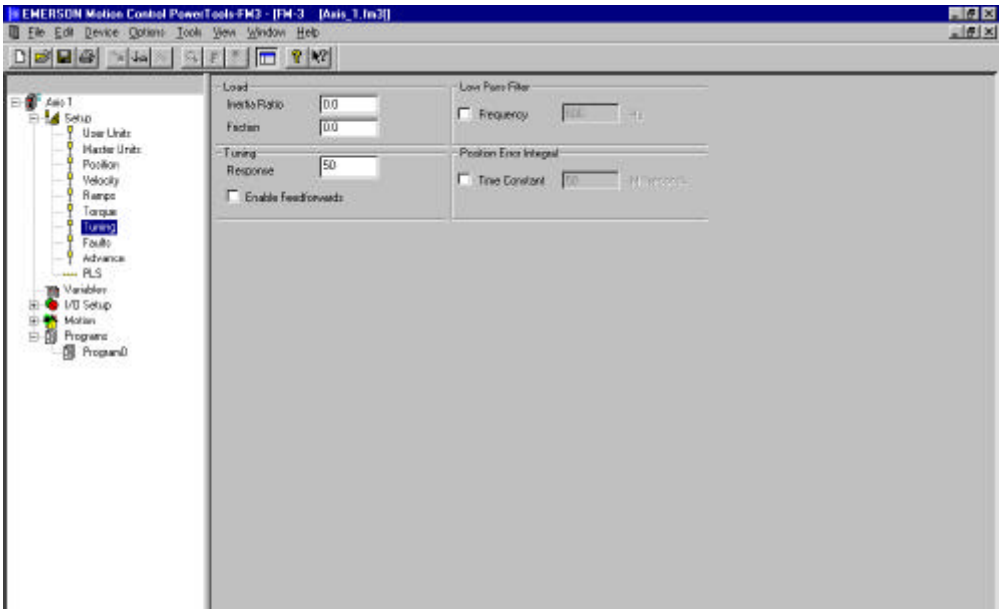


Figure 35: Tuning View

Load Group

Inertia Ratio

Inertia Ratio specifies the load to motor inertia ratio and has a range of 0.0 to 50.0. If the exact inertia is unknown, a conservative approximate value should be used. If you enter an inertia value higher than the actual inertia, the resultant motor response will tend to be more oscillatory.

Friction

This parameter is characterized in terms of the rate of friction increase per 100 motor RPM. If estimated, always use a conservative (less than or equal to actual) estimate. If the friction is completely unknown, a value of zero should be used. A typical value used here is less than one percent.

Low Pass Filter Group

The Low Pass Filter will reduce machine resonance due to mechanical coupling and other flexible drive/load components by filtering the command generated by the velocity loop.

Low Pass Filter Enable Checkbox

This enables a Low Pass Filter applied to the output of the velocity command before the torque compensator. The Low Pass Filter is only active in Pulse and Velocity modes, not in Torque mode.

Low Pass Frequency

This parameter defines the Low Pass Filter cut-off frequency. Signals exceeding this frequency will be filtered at a rate of 40 dB per decade. The default value is 600Hz.

Tuning Group

Response

The Response adjusts the velocity loop bandwidth with a range of 1 to 500 Hertz. In general, it affects how quickly the drive will respond to commands, load disturbances and velocity corrections. A good value to start with (the default) is 50 Hz. The maximum value recommended is 80 Hz.

Enable Feedforwards Check Box

When feedforwards are enabled, the accuracy of the Inertia and Friction parameters is very important. If the Inertia parameter is larger than the actual inertia, the result could be a significant overshoot during ramping. If the Inertia parameter is smaller than the actual inertia, following error during ramping will be reduced but not eliminated. If the Friction parameter is greater than the actual friction, it may result in velocity error or instability. If the Friction parameter is less than the actual friction, velocity error will be reduced but not eliminated.

Position Error Integral Group

Time Constant Check Box

This enables the Time Constant parameter.

Time Constant

Position Error Integral is a control term, which can be used to compensate for the continuous torque required to hold a vertical load against gravity. It is also useful in low speed applications, which have high friction.

The user configures this control term using the “Position Error Integral Time Constant” parameter. This parameter determines how quickly the drive will correct for in-position following error. The time constant is in milliseconds and defines how long it will take to decrease the following error to 37 percent of the original value. In certain circumstances the value actually used by the drive will be greater than the value specified here.

Min Time Constant = 1000/Response

For example, with “Response” set to 50, the minimum time constant value is 1000/50 = 20 msec.

Faults View

The Faults View displays any active faults.

See Figure 36. The Faults tab has been selected in the hierarchy view to view the Faults view.

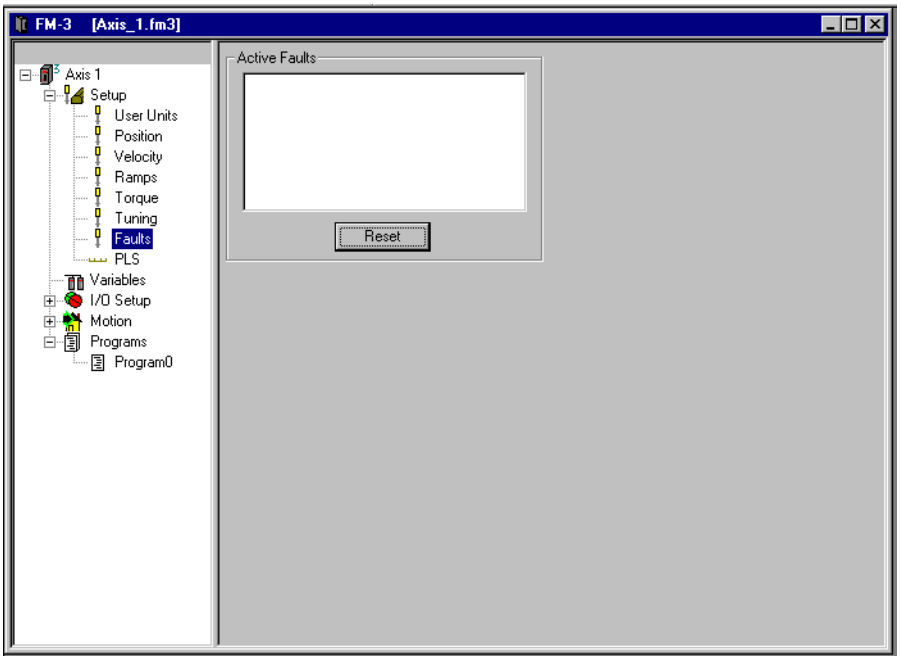


Figure 36: Faults View

Active Faults Group

The Active Faults group contains the Active Faults window.

Active Faults Window

This window displays any active faults in the system. Those faults which do not require a reboot can be cleared by clicking on the Reset button. For more detailed fault information, refer to “Diagnostics and Troubleshooting” on page 169.

PLS View

The PLS View allows users to define Programmable Limit Switches (PLS) for advanced machine operation.

By selecting PLS in the hierarchy view, the PLS View will appear in the right (see Figure 37).

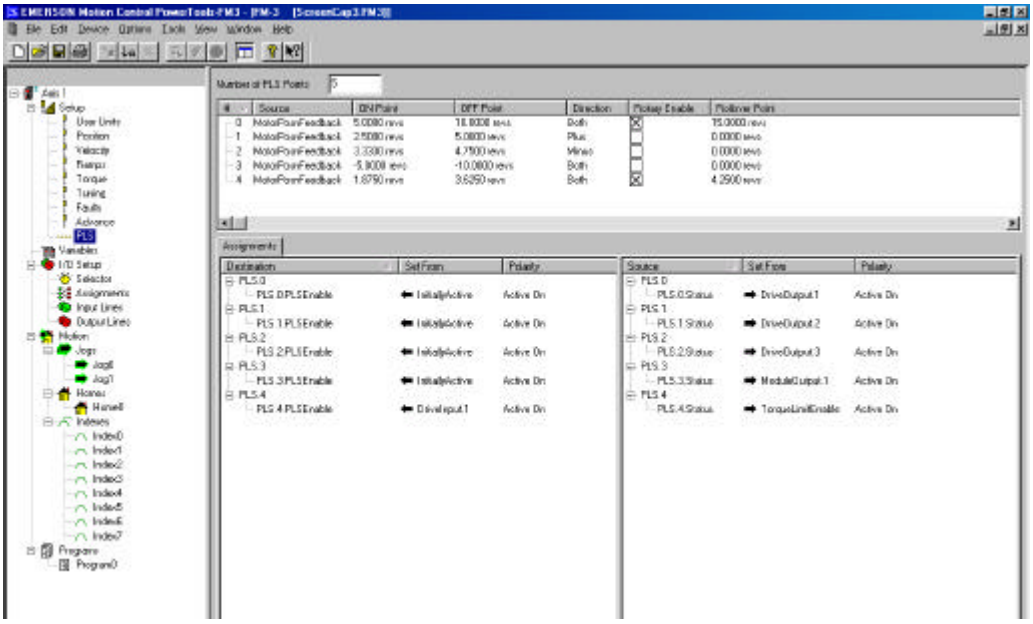


Figure 37: PLS View

A PLS can be used to turn on or off a bit based on feedback position, commanded position, or master feedback position. Eight global PLS's are available for a single application. To operate a PLS, first it must be enabled (see the PLS enable destinations in the assignments view) and then the Absolute Position Valid source must be active. Each PLS has its own OnPoint and Off Point, as well as a Rollover Point.

The terms OnPoint and Off Point assume movement in the positive direction. Those labels should be reversed if traveling in the negative direction.

Number of PLS Points

This parameter determines the number of PLS Points that will be used. Count always begins with 0, so 5 points will be 0 to 4. Up to eight PLS points may be used simultaneously.

Source

The source of a PLS can be assigned to the motor axis (MotorPosnFeedback, MotorPosnCommand) or a master synchronization signal (MasterPosnFeedback). The term motor axis refers to the motor being controlled by the FM-3/drive combination. The source list box is used to select the source for the individual PLS.

On Point

PLS.#.Status will be active when the selected source position is between the PLS.#.OnPosn and the PLS.#.OffPosn. Assume that the PLS.#.Direction is set to "Both". When traveling in the positive direction and the feedback position executes the OnPosn, the PLS.#.Status will activate. As the motor continues in the same direction, the PLS.#.Status will deactivate when feedback position reaches or exceeds the OffPosn. If motor travel changes to the negative direction, the PLS.#.Status will activate when the feedback position reaches the OffPosn, and will deactivate when it continues past the OnPosn. All on/off positions are defined in user units.

PLS.#.Status will be active if:PLS.#.OnPosn < Feedback Position ≤ PLS.#.OffPosn

Off Point

PLS.#.Status will be active when the selected source position is between the PLS.#.OnPosn and the PLS.#.OffPosn. Assume that the PLS.#.Direction is set to "Both". When traveling in the positive direction and the feedback position reaches the OnPosn, the PLS.#.Status will activate. As the motor continues in the same direction, the PLS.#.Status will deactivate when feedback position reaches or exceeds the OffPosn. If motor travel changes to the negative direction, the PLS.#.Status will activate when feedback position reaches the OffPosn, and will deactivate when it continues past the OnPosn.

PLS.#.Status will be active if:PLS.#.OnPosn < Feedback Position ≤ PLS.#.OffPosn

If using negative values for your OnPosn and OffPosn, the most negative value should go in the OnPosn parameter, and the least negative value should go in the OffPosn.

If the PLS has a rollover point, and the OnPosn is greater than the OffPosn, the PLS will be active whenever the position feedback is not between the On and Off positions, and in-active whenever the position feedback is between the two positions. However, the PLS.#.Status will not turn on until it reaches the OnPosn the first time.All on/off positions are defined in user units.

Direction

This parameter specifies the direction of motion that a particular PLS output will function. If set to Both, the PLS will activate regardless of whether the motor (or master motor) is moving in the positive or negative direction. If set to Plus, the PLS will activate only when the motor is moving in the positive direction. If set to Minus, the PLS will activate only when the motor is moving in the negative direction.

For example, a flying cutoff or flying shear application may use this feature to activate the PLS to fire the knife only when the axis is moving in the positive direction.

Rotary Enable

This parameter is used to enable the RotaryRolloverPosn for this PLS.

Rollover Point

This parameter is the absolute position of the first repeat position for this PLS. When enabled it causes the PLS to repeat every time this distance is passed. The repeating range begins at an absolute position of zero and ends at the RotaryRolloverPosn.

For example, in a rotary application a PLS could be setup with an OnPosn of 90 degrees and an OffPosn of 100 degrees. If the RotaryRolloverPosition is set to 360 degrees the PLS would come on at 90, go off at 100, go on at 450 (360+90), go off at 460 (360+100), go on at 810 (2*360+90), go off at 820 (2*360+100), and continue repeating every 360 degrees forever.

I/O Setup Group

The I/O Setup group contains four views that control input and output functions as well as other drive functions. These views are as follows: Selector, Assignments, Input Lines and Output Lines. These can be viewed by expanding I/O Setup by simply clicking on any one of the setup views underneath the I/O Setup.

Selector View

The selector View is located under I/O Setup in the hierarchy on the left of the view.

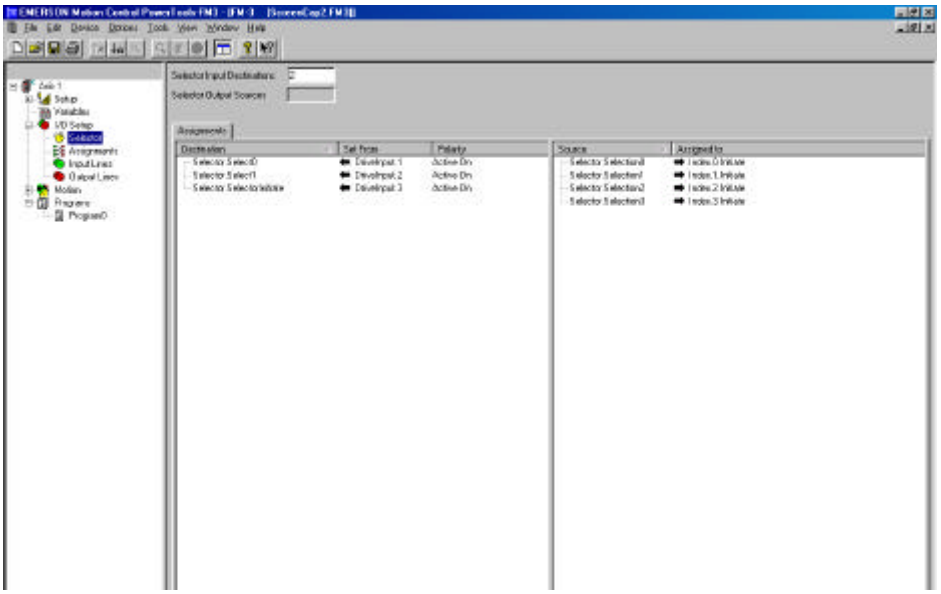


Figure 38: Selector View

The selector allows conservation of the number of input lines by using a binary input conversion to decimal. The binary select lines are set up by assigning sources to the selector.select destinations on the Assignments view. In most cases, hardware inputs are assigned to the selector.select functions (see Figure 38).

Based on the status of the binary select lines, a selector.selection source will be active when the Selector.SelectorInitiate destination is activated.

At the top of the Selector view, the Selector Input Destinations field defines how many binary select lines will be used. The number of Selector.Selections is a direct result of the number of select lines. The formula is as follows:

# of selections = 2^n where n is the number of select lines.

The maximum number of select lines is eight.

Once you have determined how many select lines you want, the assignments to these Selector.Select lines must then be made in the Assignments view.

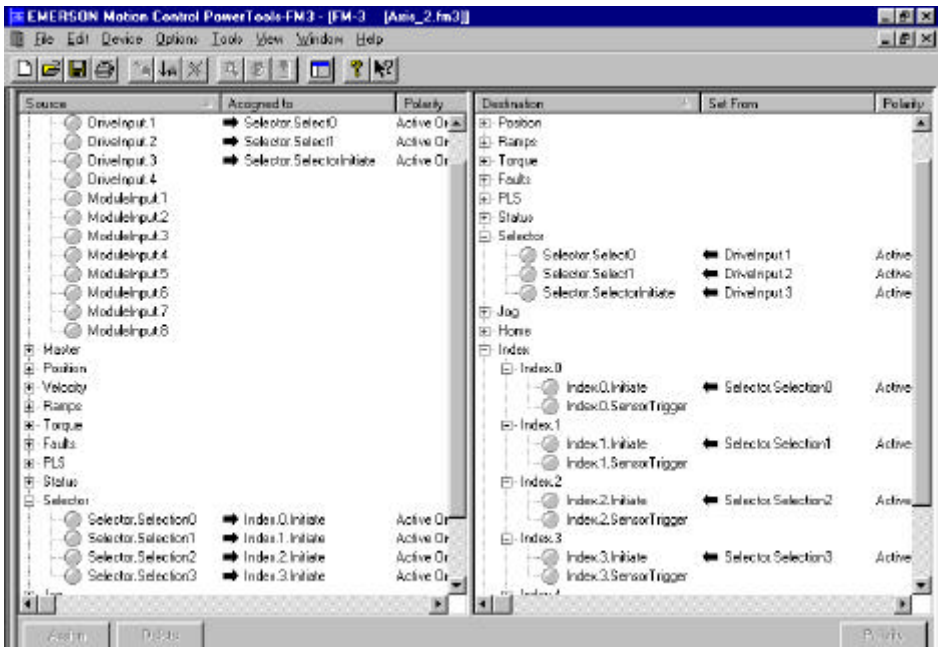


Figure 39: Assignment View

For example, if we entered 2 for the number of Input Destinations, we would have 4 selection lines (Selector.Selection0 through Selector.Selection3). The Selector.Selection number that is activated is determined by the status of the Selector.Select lines when the Selector.Selector Initiate bit is activated. Each select line has a specific binary value.

The binary value is determined as follows:

$S_n \times 2^n$ where S_n = Status of Selector.Select line n

Sn = 0 if Selector.Select line n is inactive, and

Sn = 1 if Selector.Select line n is active

The sum of all the binary values determines which Selector.Selection line will be active.

The following examples demonstrate how to determine which Selector.Selection will activate based on the Selector.Select lines.

Example 1:

If Selector.Select2 is active, Selector.Select1 is inactive, and Selector.Select0 is active, then the total binary value is as follows:

$S_2 = 1$, $S_1 = 0$, and $S_0 = 1$. Therefore,

Total Binary Value = $(1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$

Total Binary Value = 4 + 0 + 1

Total Binary Value = 5

Therefore, when Selector.SelectorInitiate activates, then Selector.Selection5 will activate.

Example 2:

If Selector.Select2 is inactive, Selector.Select1 is active, and Selector.Select0 is active, then the total binary value would be as follows:

$S_2 = 0$, $S_1 = 1$, and $S_0 = 1$. Therefore,

Total Binary Value = $(0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$

Total Binary Value = 0 + 2 + 1

Total Binary Value = 3

Therefore, Selector.Selection3 would activate.

The Selector.Select lines can change without any action until the Selector.SelectorInitiate destination is activated.

Selector.Selection sources can be tied to any destination in the Assignments view. Figure 39 shows the four selection lines being tied to Index 0 through Index 3 initiates. By doing this, we could initiate up to four indexes with only two select lines and a selector initiate. This can help minimize the number of inputs required to initiate a large number of indexes or programs.

Assignments

External control capability is provided through the use of assignments to the sources (Drive Inputs and Module Inputs) or the destinations (Drive Outputs and Module Outputs). Assignments provide a mechanism for the user to define the internal and external dynamic control structure to separate complex motion profiles. These functions directly correspond to any input or output line on the drive or the FM-3. External controllers, such as a PLC or other motion controllers, may be connected to affect or monitor the device's operation.

E Series drives are equipped with five optically isolated input lines (one dedicated to a Drive Enable function) and three optically isolated output lines. The FM-3 has an additional eight input and four output lines.

The E Series drive's input and output lines can be accessed through the removable 10-pin I/O connector (J6), or through the 44-pin command connector (J5). The FM-3 input and output lines are located on the front of the FM-3 module.

All inputs and outputs are configured as sourcing and are designed to operate from a +10 to 30 VDC power source. You are responsible for limiting the output current to less than 200 mA for each digital output.

Assignments View

The Assignments View not only displays information but also makes assignments regarding the source and the destination.

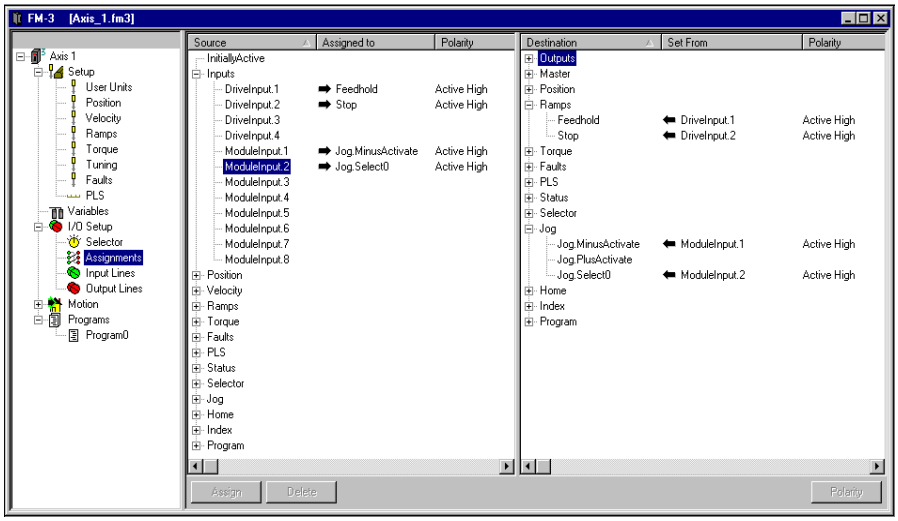


Figure 40: Assignments View

The Assignments View is used to tie a source to a destination. Destinations are functions that need to be triggered, such as Index Initiates, Program Initiates, Jog Initiates and so on.

Sources are located on the left side of the Assignment View and reflect events that occur in the drive. These events are based on drive activity. By expanding individual groups, you will see more detailed parameters. For example, if you expand the Inputs source group, you will see DriveInput.1 through ModuleInput.8, as shown in Figure 40. You can use these events to trigger certain actions (or destinations) on the right side of the view.

To make an assignment in the FM-3 module occur, you must tie a source to a destination. Any source can be tied to any destination to create the desired system operation.

Creating An Assignment:

Various methods will tie a source (such as DriveInput.1) to a destination, such as Index.0.Initiate:

Drag and Drop Method

First, position your pointer over the source on the left that you wish to assign to the destination on the right. Press the left mouse button while over the source, and hold the button down. While holding the left button down, drag the source over until your pointer is positioned over the desired destination and release the left mouse button.

Destinations can also be dragged over to sources.

Assign Button Method

Click on both the source and destination you wish to assign to each other. Once both are highlighted, the Assign button in the lower left corner of the view will become enabled. Click on the Assign button to complete the assignment. Figure 41 shows the source and destination highlighted, and the Assign button has just been clicked.

Once an assignment has been made, you will see the “Assigned to..” and the “Set From” columns be filled in for the specific sources and destinations. This indicates what destination(s) an individual source has been assigned to, and what source(s) an individual destination is assigned to.

Any source can be assigned to up to three different destinations maximum. Any destination can have as many sources as desired assigned to it.

Deleting An Assignment

Delete Button Method

Click on both the source and destination your wish to delete. Once both are highlighted, the Delete button will become available. Click on the Delete button to remove the assignment.

Right Click Method

Position the pointer over the specific assignment you with to delete and click the right mouse button. A selection box will appear. From this selection box, choose Delete.

After either of these procedures, the assignment will disappear.

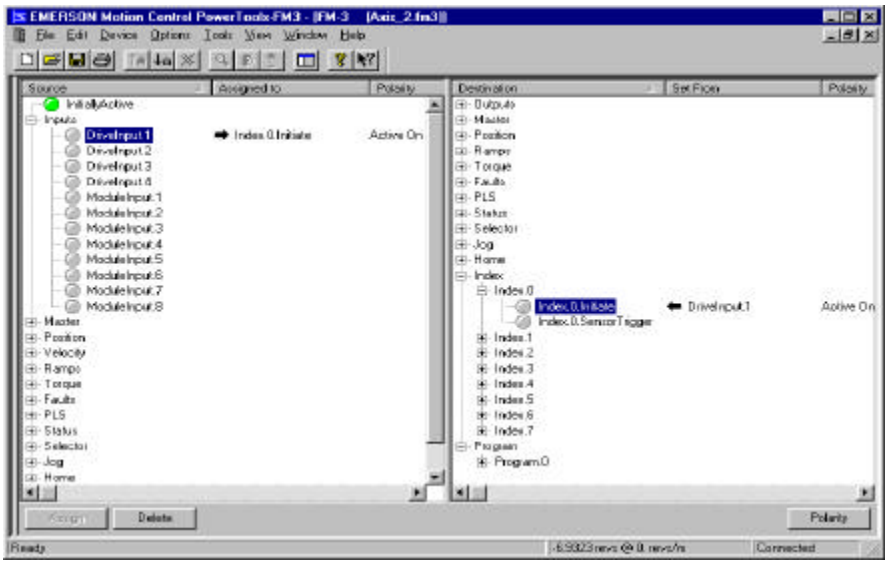


Figure 41: Tying a Source to a Destination

Assignment Polarity

The active state of an assignment can be programmed to be Active Off, Active On, or Custom using PowerTools FM-3. Making an assignment “Active On” means that the destination will be active when the source it is assigned to becomes active, and is inactive when the source is inactive. Making an assignment “Active Off” means that the destination will be active when the source it is assigned to is inactive, and will be inactive when the source is active.

The polarity of the assignment can also be changed to Custom when required. Custom polarity allows you to make a destination activate and deactivate based on two different sources.

Note

Destination functions which initiate motion (Jog.PlusInitiate, Jog.MinusInitiate, Index.#.Initiate, and Home.#.Initiate) cannot be set “Active Off”.

Default polarity for a new assignment is Active On. Two methods will change the polarity of an assignment.

Polarity Button Method

Click on both the source and the destination to be changed. Once highlighted, the Polarity button will become available in the lower right corner of the view. Click on the Polarity button and change the settings as desired in the Polarity edit box. Click OK to apply the changes.

Right Click Method

Position the pointer over the specific assignment you with to change polarity for and click the right mouse button. A selection box will appear. From this selection box, choose Polarity. A Polarity edit box will appear. Change the Polarity settings as desired and click OK to apply the changes.

User Level

The User Level filters the available assignments. The User Level is changed on the Options menu at the top of the PowerTools toolbar. Choose Options/Preferences/User Levels. Easy mode filters out all but the most commonly used sources and destinations. Detailed mode filters out less, expanding the list of sources and destinations for more complex configurations. Too Much mode does not filter at all and provides all sources and destinations.

Input Lines View

The Input Lines View displays any functions that have been assigned to the drive or module hardware inputs. See Figure 42.

Note

No assignments can be made using the Input Lines View, assignments are only displayed in the Input Lines View.

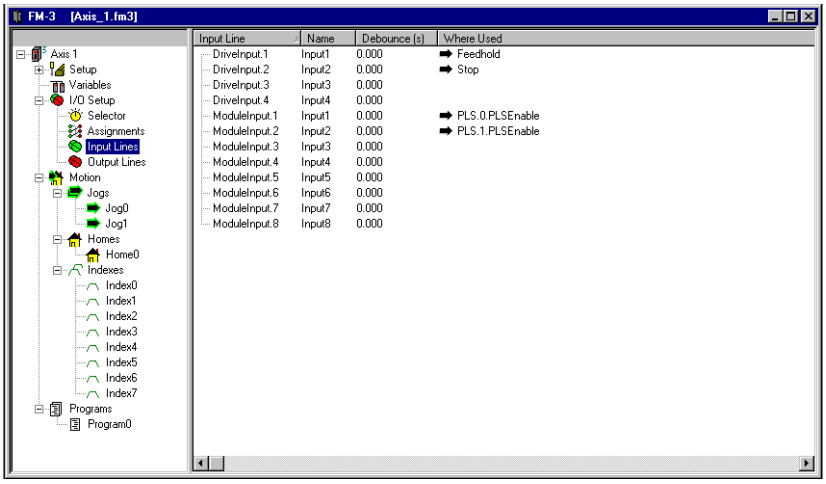


Figure 42: Input Lines View

Two functions can be performed on the Input Lines view.

Name

You can assign a descriptive name to each input and make the setup easier to follow. The length of the text string is limited by the column width with a maximum of 12 characters. Simply double click on the Name field of any input line to assign a name to it.

Debounce

You can program a “Debounce Time” to any input line, which means the motion profile will need to be steady for at least the debounce time before it is recognized. This feature helps prevent false triggering in applications in noisy electrical environments. At the end of the debounce time, the next action can occur.

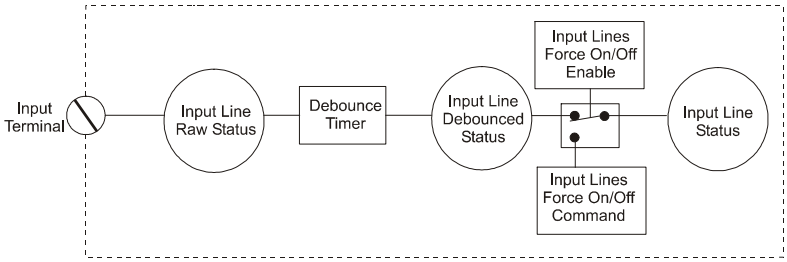


Figure 43: Input Line Diagram

If the Input Line attached to the home sensor is debounced, the actual rising edge of the Home Sensor is used to determine the Home Reference Position (the debounce time ensures a minimum pulse width).

Output Lines View

The Output Lines View displays any functions that have been assigned to the drive or module hardware outputs. See Figure 44.

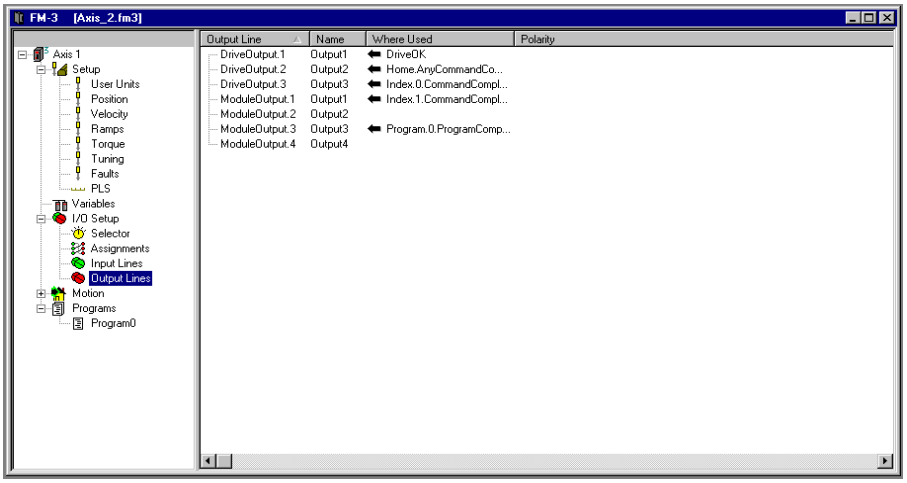


Figure 44: Output Lines View

Names

Descriptive text names can be assigned to individual output lines to make the setup easier to follow.

Motion Group

All motion parameters related to Jogs, Homes, and Indexes are located in the Motion hierarchy group.

Motion views will use units that correspond to Realtime or Synchronized motion. This choice is made on each motion view. The units are customized in the Setup Group: Realtime units are defined on the User Units View, and Synchronized units are defined on both the User Units View and the Master Units View.

Each of the motion views, when online, have an Online tab that displays feedback information and provides buttons to initiate motion.

Jog View

Jogging produces rotation of the motor at controlled velocities in a positive or negative direction. The jog is initiated with the Jog.#.Initiate destination or from a program.

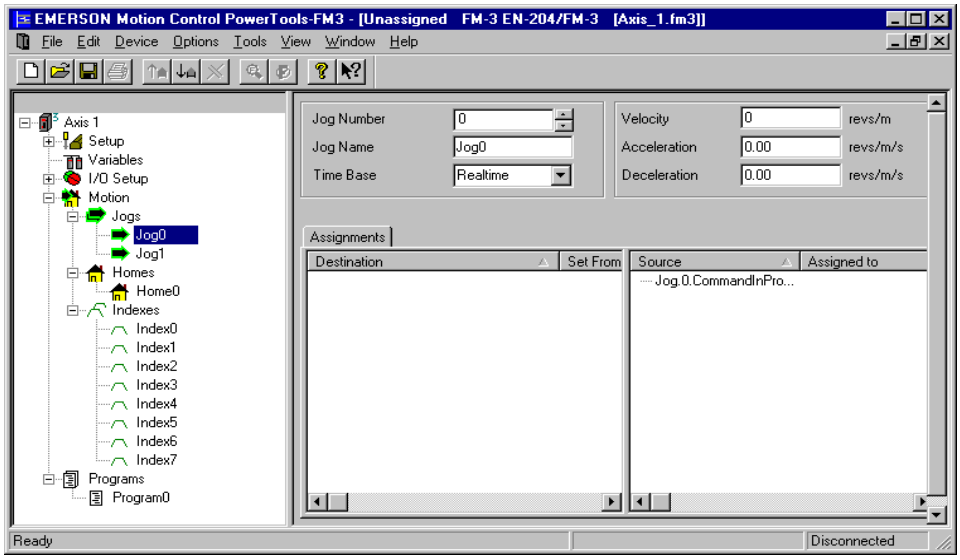


Figure 45: Jog Tab

Jog Number

This scroll box allows you to select between Jog0 and Jog1 setup views.

Jog Name

This is a descriptive character string which can be assigned to the specific jog. Giving a name to a jog can make the motion setup easier to follow.

Time Base

This list box allows the user to select the time base for the individual jog. The options are Realtime and Synchronized.

Jog Velocity

This parameter specifies the target jog velocity for the individual Jog. The motor will run at this velocity when jogging with an assignment or through a program. This value is always positive. The direction of the jog is determined by using the Jog.PlusInitiate or the Jog.MinusInitiate, which jog in the positive or negative directions respectively.

Jog Acceleration

This is the acceleration ramp used when initiating this individual Jog. If S-Curve ramps are used, then this is the average acceleration rate for the entire ramp. The units for the acceleration are setup in the Setup - User Units view in PowerTools FM-3.

Jog Deceleration

This is the deceleration ramp used when stopping this individual Jog. If S-Curve ramps are used, then this is the average deceleration rate for the entire ramp. The units for the deceleration are setup in the Setup - User Units view in PowerTools FM-3.

Jog Sources and Destinations

Sources

Jog.AnyCommandComplete

The Jog.AnyCommandComplete source will activate when either Jog0 or Jog1 completes its deceleration ramp, and reaches zero commanded velocity. It will deactivate when any Jog is initiated again. If the Stop destination is used during a Jog, then the Jog.AnyCommandComplete will not activate.

Jog.#.Accelerating

This source is active while a jog is accelerating to its target velocity. Once the Jog reaches the target velocity, the Jog.#.Accelerating source will deactivate.

Jog.#.AtVel

This source activates when the individual jog reaches its target velocity. It deactivates when a jog deceleration ramp begins.

Jog.#.CommandInProgress

The Jog.#.CommandInProgress source is active throughout an entire jog profile. The source activates at the beginning of a jog acceleration ramp, and deactivates at the end of a jog deceleration ramp.

Jog.#.CommandComplete

The Jog.#.CommandComplete source will activate when the specific jog completes its deceleration ramp. It will remain active until the specific jog is initiated again. If the Stop destination is used during a Jog, then the Jog.#.CommandComplete will not activate.

Jog.#.Decelerating

This source is active while a jog is decelerating from its' target velocity. Once the Jog reaches zero velocity (or its new target velocity), the Jog.#.Decelerating source will deactivate.

Destinations

The following destination functions can be found in the Assignments view under the I/O setup group:

Jog.PlusActivate

When this destination is activated, jogging motion will begin in the positive direction. The jog velocity is determined by whether the specified jog (specified by Jog.Select0) is active or not. A jog stops when this destination is deactivated.

Jog.MinusActivate

When this destination is activated, jogging motion will begin in the negative direction. The jog velocity is determined by whether the specified jog (specified by Jog.Select0) is active or not. A jog stops when this destination is deactivated.

Jog.Select0

This destination is used to select between Jog0 and Jog1. When the Jog.Select0 destination is not active, the target velocity for the jog is the Jog.0.Velocity. If the Jog.Select0 destination is active, the target velocity of the jog is the Jog.1.Velocity. Jog.Select0 can be toggled "On" or "Off" while jogging. Jog acceleration and deceleration ramps are used to ramp between jog velocities.

Below is a description of jog operation using these destinations.

Note

In the table below Jog.0.Velocity = 100 RPM and Jog.1.Velocity = 500 RPM.

Jog.PlusActivate	Jog.MinusActivate	Jog.Select0	Motion
Off	Off	Off	0 RPM
On	Off	Off	+100 RPM
Off	On	Off	-100 RPM
On	Off	On	+500 RPM
Off	On	On	-500 RPM
On	On	Off	0 RPM
On	On	On	0 RPM

All Jog destinations are level sensitive.

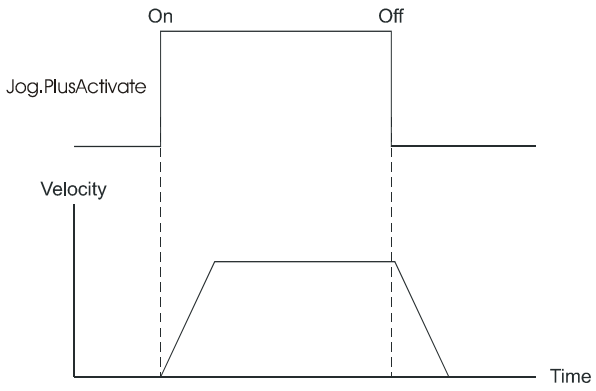


Figure 46: Jog Activation

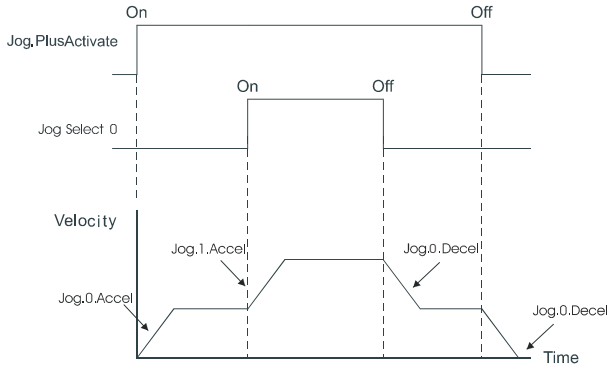


Figure 47: Jog Select Details

If the Jog direction is reversed, the Jog.#.Decel value will be used to decelerate the motor to zero speed and then the Jog.#.Accel will be used to accelerate to the new (opposite sign) velocity.

Note

The Jog destinations cannot be initiated when any other motion type (homing, indexing, or programs) is in progress.

If both jog input functions are “On” there is no motion after a jog deceleration (they effectively cancel each other). The drive’s display will show “R”, for ready.

If the device is jogging with the Jog.PlusActivate destination active and the Jog.MinusActivate destination activates, the motor will behave the same as if it would if Jog.PlusActivate just deactivated.

The Stop destination (found under the Ramps group in the Assignments view) will override the Jog operation and decelerate the motor to zero speed at the stop deceleration rate.

If the motor reaches a Travel Limit, you can Jog off the Travel Limit in the opposite direction. (Use Jog.PlusActivate to move off a Travel Limit -).

Home View

The Home is used in applications in which the axis must be precisely aligned with some part of the machine. The Home is initiated with the Home.#.Initiate Destination or from a program.

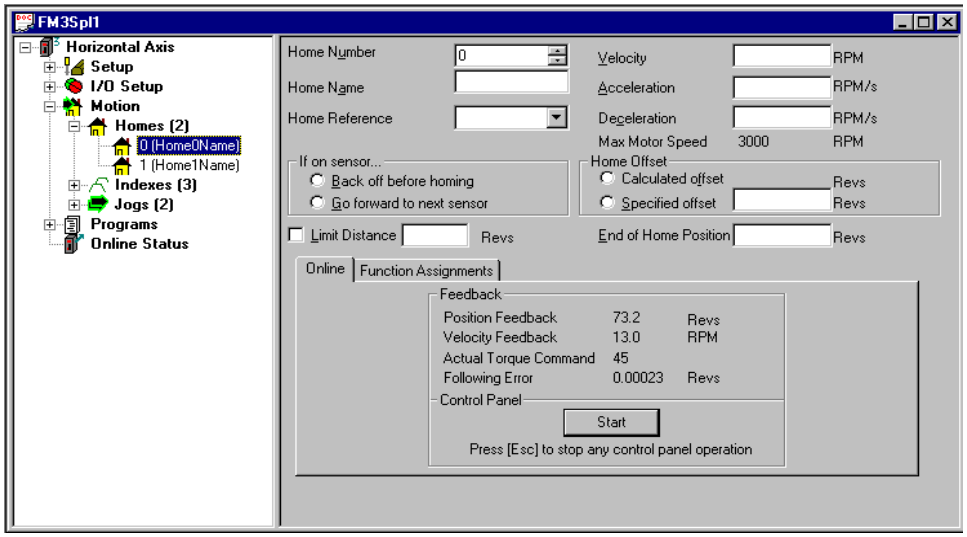


Figure 48: Home View, Online

Home Number

The Home Number parameter displays which home sequence you are editing and allows you to scroll through multiple home sequences using the up and down arrows. The first release only allows for one home sequence.

Name

Allows you to assign a descriptive name to the home sequence up to 10 characters in length.

Home Reference

This parameter determines the signal used as the reference. The parameter can have one of three different values: 'Sensor', 'Marker', or 'Marker then Sensor'. When the home reference is 'Sensor' the rising edge of the 'Home.#.SensorTrigger' destination is used to establish the home position. When the home reference is 'Marker' the rising edge of the motor encoder's marker channel is used to establish the home position. When the home reference is 'Sensor then Marker' the home position is established using the first marker rising edge after the Home.#.SensorTrigger destination activates.

Time Base

Selects the Time Base for the home move velocity and acceleration/deceleration. Real-time and sync are the allowed selections.

Velocity

Sets the target velocity for the home. The polarity determines the home direction. Positive numbers cause motion in the positive direction and negative numbers cause motion in the negative direction in search of the home reference.

Acceleration

Average Acceleration rate used during the home. Units are specified on the User Units page.

Deceleration

This is the average Deceleration rate used at the end of the Home move in user units.

If on sensor... Group

These radio buttons determine how the system reacts if the Home.#.SensorTrigger is already active when the home is initiated.

'Back off before homing' Radio Button

If this radio button is set, the drive will back off the sensor before beginning the home. It does this by moving the direction opposite to that specified by the sign of the home velocity. It continues moving in this direction at the target home velocity until the sensor goes deactivates. The motor then decelerates to a stop and performs a standard home.

'Go forward to next sensor' Radio Button

If this radio button is set, then the system will ignore the sensor that is active when the home is initiated, and move in the proper direction until the first low to high transition of the Home Reference signal.

Home Offset Group

The Home Offset group has two buttons, the calculated Offset Radio Button and the Specified Offset Radio Button.

Calculated Offset Radio Button

The calculated offset is defined as the distance traveled during deceleration ramp from the home velocity to a stop plus the distance traveled at the home velocity for 1600µs. This extra distance is used to guarantee that the motor will not need to backup after the deceleration ramp.

Specified Offset Radio Button

The specified offset allows the user to choose an exact offset from the Home Reference point. The commanded motion will stop at exactly the offset distance away from the reference point

as specified. If the specified offset is smaller than the calculated offset, the motor will decelerate to a stop and then back up to its final offset position.

Limit Distance

LimitDistEnable

This checkbox enables the specified Home Limit Distance.

The Limit Distance parameter places an upper limit on the incremental distance traveled during a home. If no home reference is found in this distance, the motor will decelerate to a stop at the limit distance and activate the Home.#.LimitDistHit source.

End of Home Position

This parameter defines the position at the completion of the home. This defaults to 0.0 such that at the end of a home, the Feedback Position and the Commanded Position are set to 0.0. If you wish your Feedback Position to be something other than 0.0 at the end of a home, then enter the exact desired position here.

Below is a diagram of a home using the "Back off before homing" radio box, a Home Reference of "Sensor", and using a "Calculated Offset".

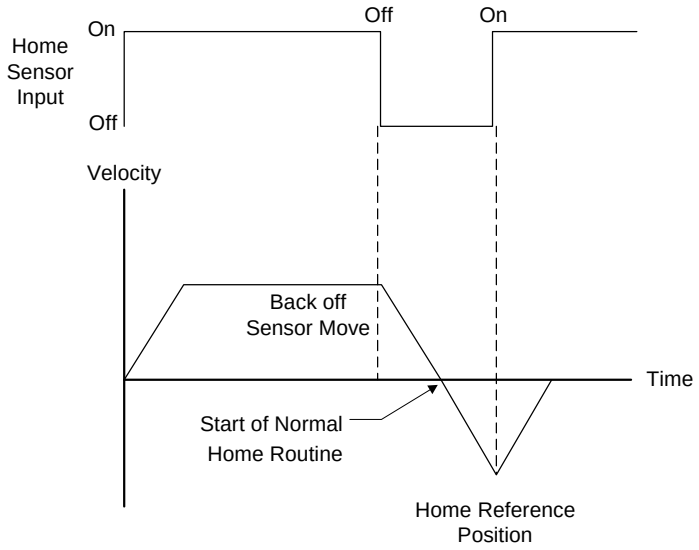


Figure 49: Home Reference Position

Home Sources and Destinations

Sources

Home.AbsolutePosnValid

This source is activated when a Home is successfully completed. It indicates that the device has been homed properly. It is will be deactivated by the Home.#.Initiate destination, an encoder fault, a reboot, or when the device is powered down.

Home.AnyCommandComplete

This source is activated when any home motion command is completed. If a drive stop destination is activated before the home has completed, this source will not activate. It will be deactivated when another home is initiated.

Home.#.Accelerating

This source is active while a home is accelerating to its target velocity. Once the home reaches the target velocity, the Home.#.Accelerating source will deactivate. This source will also activate during the "back off sensor" motion before the actual home.

Home.#.AtVel

This source activates when the home reaches its target velocity. It deactivates when a home deceleration ramp begins. Home.#.AtVel will not be activated during the "back off sensor" portion of the home.

Home.#.CommandComplete

The Home.#.CommandComplete source will activate when the specific home completes its deceleration ramp. It will remain active until the specific home is initiated again. If the drive stop destination is used during a home, then the Home.#.CommandComplete will not activate.

Home.#.CommandInProgress

Activated when the Home is initiated and remains active until all motion related to the Home has completed.

Home.#.Decelerating

This source is active while a home is decelerating from its target velocity. Once the home reaches zero velocity (or its' new target velocity), the Home.#.Decelerating source will deactivate. This source will also activate during the "back off sensor" motion before the actual home.

Home.#.LimitDistHit

This source is activated when the home reference is not found before the Home Limit Distance is traveled. It will remain active until the home is initiated again.

Destinations

Home.#.Initiate

The Home.#.Initiate destination is used to initiate the home function. The Home is initiated on the rising edge of this function. The device will not initiate a Home if there is an Index, Jog, or Program in progress, or if the Stop destination is active or if a travel limit is active.

Home.#.SensorTrigger

This destination is required to be used if you are homing to a sensor. This destination is edge sensitive. The home position is determined when the Home Sensor destination is activated.

If the device receives a Home.#.Initiate input while the Home.#.SensorTrigger is active, you can choose to have the motor “back-off” of the home sensor before it initiates the home function, or move forward to the next sensor.

If debounce is used on the hardware input that the Home.#.SensorTrigger is assigned to, the debounce determines the length of time the input must be active to be considered a valid input. The rising edge of the sensor is still used for the reference position. This maintains accuracy while providing the ability to ignore false inputs.

Index View

An index is a complete motion sequence that moves the motor a specific incremental distance or to an absolute position. The index is initiated with the Index.#.Initiate destination or from a program.

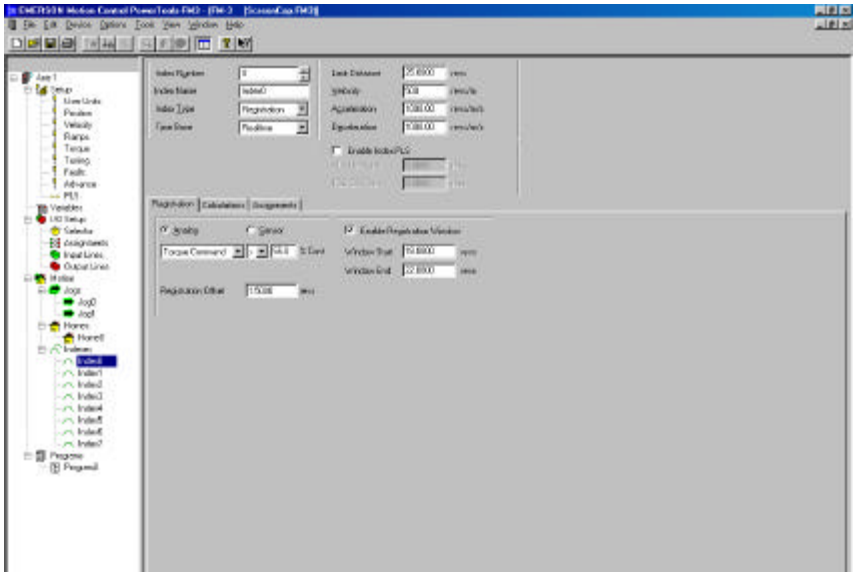


Figure 50: Index View

Index Number

The Index Number parameter selects the index number with a scroll box.

Index Name

The User can specify an Index name of up to 12 alphanumeric characters. This allows assigning a descriptive name to each index indicating different machine operations.

IndexType

Select the index type from Incremental, Absolute, Registration, Rotary Plus, or Rotary Minus. Click the down arrow on the parameter list box to select the desired type of Index profiles, as follows:

Incremental Indexes run a specified distance from the current position.

Absolute Indexes move to an exact position with respect to the home reference point. The absolute index could run in either a clockwise or counterclockwise direction dependent on the current position when it is initiated.

A Registration Index runs at the specified velocity until a registration sensor is seen or until it reaches the Registration Limit Distance. If a Registration Sensor is seen, then the index runs an additional Registration Offset distance.

Rotary Plus and Rotary Minus type indexes are typically used in applications which use rotary rollover. These absolute indexes are forced to run in a specific direction regardless of the starting point.

TimeBase

This list box selects the Time Base for the index velocity and acceleration/deceleration. Real-time and sync are the allowed selections.

Distance/Position

The Distance/Position parameter is a signed value which specifies the distance the index will travel (incremental index) or the absolute position the index will move to (absolute index). In the case of an incremental index, this parameter also determines the direction the index will travel. If an index type of Registration is selected, then this is a limit distance, or the maximum distance the index will travel if a registration sensor is not seen.

Velocity

This sets the target velocity for the index profile. The velocity parameter is unsigned and must be greater than zero. Direction of the index is not determined by the velocity, but by the Distance/Position parameter.

Acceleration

Average Acceleration rate used during the index. Units are specified on the User Units page.

Deceleration

The Deceleration parameter specifies the deceleration value to be used during the index in user units.

Enable Index PLS

This check box enables (when checked) or disables the Index PLS function.

An Index PLS is similar to a global PLS (explained in the PLS View section), but is incremental in nature. The Index PLS has On and Off points just like a global PLS, but the On and Off points are specified as an incremental distance from the start of the index, instead of absolute positions. Each index has its' own On and Off points, and the Index.#.PLSStatus is only updated when Index# is run. The direction of the PLS does not matter, the Index.#.PLSStatus will activate and deactivate the same incremental distance from the start of the index.

PLS On Point

This parameter is an incremental distance from the start position of the index, at which the PLS.#.Status will become active. It is an unsigned value in user units. The On Point must always be less than the Off Point.

PLS Off Point

This parameter is an incremental distance from the start position of the index, at which the PLS.#.Status will deactivate. It is an unsigned value in user units. The Off Point must always be greater than the On Point. If the Off Point is larger than the Distance parameter in an Incremental type of index, the PLS Status will never deactivate until the index is run again.

Example 1:

Index 0 is an Incremental index with a distance of 5 Revs. The PLS On Point is set to 1 Rev, and the PLS Off Point is set to 4 Revs. A home is completed, and Position Feedback is equal to 0.0 Revs.

If Index 0 is run, the Index.0.PLSStatus will activate when the feedback position reaches 1 Rev and remain active until feedback position reaches 4 Revs, and deactivate. At the end of Index 0, position feedback is equal to 5 Revs. If we initiate Index 0 again, Index.0.PLSStatus will activate 1 Rev into the index, or at 6 Revs. It will remain active until position feedback reaches 9 Revs, and deactivate. This index could be run over and over again, and Index.0.PLSStatus will activate 1 Rev from the starting position and deactivate 4 Revs from the starting position every time.

Example 2:

Index 1 is an Incremental index with a distance of -10 revs. The PLS On Point is set to 4 Revs, and the PLS Off Point is set to 6 Revs. A home is completed, and Position Feedback is equal to 0.0 Revs.

If Index 1 is run, the Index.1.PLSStatus will activate when the position feedback reaches -4 Revs (or 4 Revs from the start of the index). Index.1.PLSStatus will then deactivate when position feedback reaches -6 Revs (or 6 Revs from the start of the index). If Index 1 is run again, Index.1.PLSStatus will activate and deactivate at -14 Revs and -16 Revs respectively.

Index PLS's can be used on any type of an index.

If an index is so short (possible in the case of an absolute index) that it reaches the On Point, or incremental distance, into the index, but never reaches the Off Point, the Index.#.PLSStatus will remain active until the index is run again.

Similarly, if the index is so short that it never reaches the On Point, the Index.#.PLSStatus will never activate.

Registration Parameters

The following parameters are only used if Registration is selected as the Index Type.

'Analog' or 'Sensor' Radio Buttons

Select one of these radio buttons to determine what signal will be used as your registration trigger.

If 'Sensor' is selected, a source must be assigned to the Index.#.SensorTrigger Typically a proximity sensor is wired to a hardware input, and therefore a module or drive input source is assigned to the Index.#.SensorTrigger, but any source can be used.

If 'Analog' is selected, one of the analog signals must be selected in the analog list box. Available selections are Torque Command or Torque Feedback. Then a comparison operator must be selected from the operator list box. Available selections are > (greater than) and < (less than). Last, an analog value must be entered for comparison.

Registration Offset

The incremental distance the motor will travel after a valid registration sensor or analog limit value has been detected.

Enable Registration Window

This check box enables (if checked) the Registration Sensor Valid Window. When active, only registration marks that occur inside the registration window are seen as valid.

Window Start

This parameter defines the start of the Registration Sensor Valid Window relative to start position of this index. This is an unsigned value and is relative only to starting position of this index. Index direction does not affect this parameter. The Registration Window Start position (or distance) should be less than the Registration Window End position. If a registration sensor is seen outside of this window (not between the WindowStart and WindowEnd positions) then it will be ignored.

Window End

This parameter defines the end of the Registration Sensor Valid Window relative to start position of this index. This is an unsigned value and is relative only to starting position of this index. Index direction does not affect this parameter. The Registration Window End position (or distance) should be greater than the Registration Window Start position. If a registration sensor is seen outside of this window (not between the WindowStart and WindowEnd positions) then it will be ignored.

Example:

Index 0 is defined as a Registration type of index. The user wants the index to run at velocity for 10 Revs, or until the Torque Feedback reaches 50% continuous torque and then continue for another 0.5 Revs.

In the Limit Distance parameter, enter 10.0

On the registration tab, select the Analog radio button.

In the analog list box, select Torque Command

In the comparison operator list box, select ">"

In the analog value parameter, enter 50 (Units are established on the User Units view)

In the Registration Offset parameter, enter 1.5

This index would accelerate up to its' target velocity, and run at speed until one of the following:

The Limit Distance is approaching, and the index decels down to zero velocity, completing the move at the Limit Distance. At this point, the Index.#.LimitDistHit source would activate. Or,

The Torque Command reaches or exceeds 50% continuous, and the index continues at speed before decelerating to zero velocity at the registration point plus the Registration Offset distance.

If the Registration Offset distance is in the opposite direction from the move, or is so short that the motor cannot stop in the specified distance at the programmed deceleration rate, the motor will decelerate with the programmed ramp, and then back-up to the specified position (registration point + the Registration Offset).

Index Sources and Destinations

Sources

Index.AnyCommandComplete

Active when any index motion command is completed. If a stop is activated before the index has completed, this destination will not activate. Deactivated when any new index command is initiated.

Index.#.Accelerating

This source is active while an index is accelerating to its' target velocity. Once the index reaches the target velocity, or begins to decelerate, the Index.#.Accelerating source will deactivate.

Index.#.AtVel

This source activates when the target index velocity is reached. If Feedrate override is changed or FeedHold is activated AtVelocity shall remain active. Index.#.AtVel will deactivate at the start of any deceleration or acceleration. During a synchronized index, this source could be active even without any motor motion if the master axis stops.

Index.#.Command Complete

The Index.#.CommandComplete source will activate when the specific index completes its deceleration ramp. It will remain active until the specific index is initiated again. If the drive stop destination is used during an Index, then the Index.#.CommandComplete will not activate.

Index.#.Command In Progress

The Index.#.CommandInProgress source is active throughout an entire index profile. The source activates at the beginning of the index acceleration ramp, and deactivates at the end of the index deceleration ramp. During a synchronized index, this source could be active even without any motor motion if the master axis stops.

Index.#.Decelerating

This source is active while an index is decelerating from its' target velocity. Once the index reaches zero velocity, or its' next target velocity, the Index.#.Decelerating source will deactivate.

Index.#.LimitDistHit

Activated when the registration sensor is not found before the Limit Distance is traveled. If the Registration Window is enabled the sensor must be activated inside the window to be recognized.

Index.#.PLSStatus

Controlled by the PLSOn and PLSOff Points which are relative to the distance commanded since the start of the index. Activated when index distance command is in between the PLSOn point and PLSOff points.

Destinations

Index.#.Initiate

The Index.#.Initiate destination is used to initiate the specific index. The Index is initiated on the rising edge of this function. An Index cannot be initiated if there is an Home, Jog, or Program in progress, or if the Stop destination or if a travel limit is active. It can be activated from an assignment or from a program.

Index.#.Sensor Trigger

If registration to Sensor is selected, when this destination activates, motor position is captured and is used as the registration point for registration type indexes.

Network Group

Modbus View

The Modbus View is used to assign Modbus addresses to individual parameters.

By selecting Modbus in the hierarchy view, the Modbus View will appear in the right (see Figure 51). The right part of the screen displays all of the drive parameters. The number of parameters that appear depends on the User Level (see the User Level section earlier in this chapter).

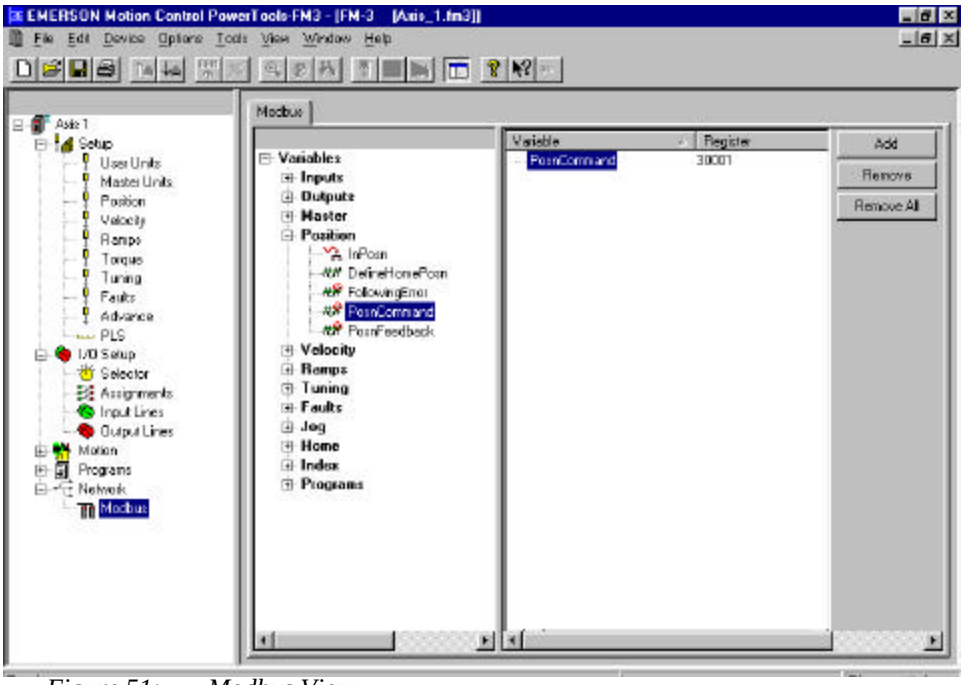
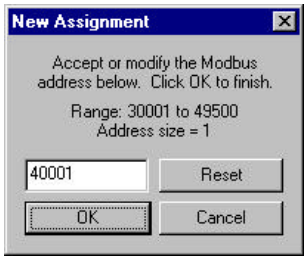


Figure 51: Modbus View

An external device such as a Human Manchine Interface (HMI) or PLC can be used to monitor or edit individual FM-3 parameters. The FM-3 and E Series drives use a 32-bit Modbus RTU communications protocol.

In order to view or modify a parameter, a Modbus address must be assigned to the specific parameter. To do this, locate the parameter you wish to read/write to or from in the variables list in the middle of the view. Once you have found the proper parameter, click and hold the left mouse button over the parameter. While still holding the button on your mouse, drag the parameter into the Modbus window area on the right of the view. Now let go of the mouse button.

A “New Assignment” dialog box will appear. This will automatically assign the next available modbus address, or allow you to enter a different Modbus address. Once you click on OK, then you will be able to read or write the parameter at that address.



Address ranges are as follows:

Address Range	Accessibility	Type	Data Size
4xxxx	Read/Write	Register	32 bit word
3xxxx	Read Only	Register	32 bit word
1xxxx	Read Only	Input Bits	bit
0xxxx	Read/Write	Coil	bit

Any individual Modbus address can be deleted by highlighting the parameter you wish to delete, and clicking on the Remove button. The address selected will be removed from the list. If you wish to delete all of the Modbus addresses that have been created, then simply click on the Remove All button. All of the addresses will disappear and the Modbus window will be empty.

Some Modbus addresses have been reserved and can not be assigned:
39980-39999 and 49501-49999.

Programming

Motion Programs are a series of indexes, homes and jogs that have been previously setup. You combine these with other programming steps to create a complex motion profile. Each motion program provides a series of movements in conjunction with other machine functions. The movements are used to perform a particular machine operation.

Multiple programs can be created using Emerson Motion Control PowerTools® FM-3 software and stored in the FM-3. The FM-3 is capable of storing up to 55 indexes, 99 motion programs, and a maximum of 1024 program steps in the FM-3 Flash Memory. The amount of available Flash Memory determines how many programs, program steps, indexes, etc. that the configuration can hold.

The number of available programs and average number of steps per program are directly related to each other. The memory is setup such that if you require 99 programs (maximum), each program can have an average of 10 program steps each. If the number of programs is reduced to a minimum, you could have as many as 1024 steps in a single program.

Program Instruction Types

Motion Instructions

Index.Initiate

This program instruction is used to initiate a single index. The index is preset to include an acceleration up to speed, a run at speed and a deceleration to a stop.

A comment is automatically inserted after the index instruction which shows key data about the particular index. The comment starts with the apostrophe ‘character.

A “Wait For Index.AnyCommandComplete” instruction is also automatically inserted after each index. This insures that the index has completed before the program continues on to the next line of code. It is also possible make the program wait until the index is complete and the following error is less than a specified amount. This is accomplished by changing the “Wait For Index.AnyCommandComplete” to “Wait For InPosn”. The In Position Window is configured in the Position view.

Examples:

```
Index.0.Initiate                ‘Incremental,Dist=5.000in,Vel=2.0in/s
Wait For Index.AnyCommandComplete

Index.37.Initiate               ‘Absolute,Posn=120.60mm,Vel=50.2mm/s
Wait For InPosn
```

Index.CompoundInitiate

This program instruction is used to initiate an index which has no deceleration ramp. The index accelerates or decelerates towards the next index velocity using the next index

acceleration ramp. The index will finish at velocity. The program then moves on to the next index. It smoothly transitions into the second index without stopping. The second index then ramps to its pre-configured velocity. Multiple indexes can be “compounded” to create a complex velocity profile. The last index in a complex profile must have a deceleration ramp. This is accomplished using a standard Index.Initiate rather than a Index.CompoundInitiate. The final index will honor the deceleration ramp. If the last index is not long enough to perform a decel ramp at the programmed rate, the motor will backup at the end of the last index.

Each index can be used in multiple places as both a standard index with a deceleration ramp, and a compound index without a deceleration ramp. The program instruction (Index.0.Initiate or Index.0.CompoundInitiate), not the index itself, determines whether or not the index will execute a deceleration ramp. For example, Index.0 can be used multiple times in multiple programs. It can be initiated at different times using the Index.0.Initiate instruction and the Index.0.CompoundInitiate instruction.

A comment is automatically inserted after the index instruction which shows key data about the particular index. The comment starts with the apostrophe ‘ character.

Examples:

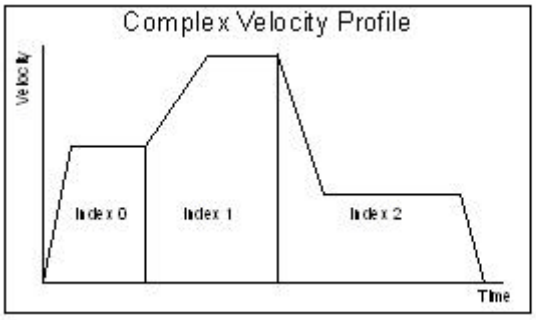


Figure 52: Index Velocity Profile

```
Index.0.CompoundInitiate            ‘Incremental,Dist=5.000in,Vel=50in/s
Index.1.CompoundInitiate            ‘Incremental,Dist=20.000in,Vel=75in/s
Index.2.Initiate                    ‘Incremental,Dist=10.000in,Vel=30in/s
Wait For Index.AnyCommandComplete
```

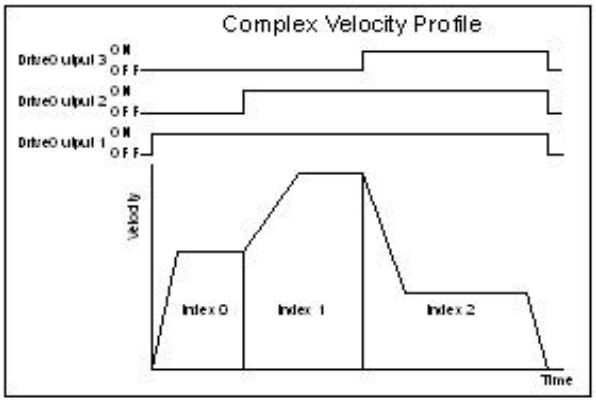


Figure 53: Index Velocity Profile with Drive Outputs

```
Index.0.CompoundInitiate ‘Incremetal,Dist=5.000in,Vel=50in/s
DriveOutput.1=ON         ‘Turns ON immediately after Index.0 is started
Index.1.CompoundInitiate ‘Incremental,Dist=20.000in,Vel=75in/s
DriveOutput.2=ON         ‘Turns ON immediately after Index.1 is started
Index.2.Initiate         ‘Incremental,Dist=10.000in,Vel=30in/s
DriveOutput.3=ON         ‘Turns ON immediately after Index.2 is started
Wait For Index.AnyCommandComplete
DriveOutput.1=OFF        ‘Turns OFF after Index.2’s command is completed
DriveOutput.2=OFF        ‘Turns OFF after Index.2’s command is completed
DriveOutput.3=OFF        ‘Turns OFF after Index.2’s command is completed
```

Home.Initiate

This program instruction is used to initiate the home.

A comment is automatically inserted after the Home.Initiate instruction which shows key data about the particular home. The comment starts with the apostrophe ‘ character.

A “Wait For Home.AnyCommandComplete” instruction is not required because the home is actually a program which already has a “Wait For” instruction.

Example:

```
Home.0.Initiate      ‘Sensor,Offset=2.000in,Vel=-10.0in/s
```

Jog.PlusInitiate

This program instruction is used to initiate jogging in the positive direction. The Jog.Stop instruction is used to stop jogging motion.

A comment is automatically inserted after the Jog.PlusInitiate instruction which shows key data about the particular jog. The comment starts with the apostrophe ‘ character.

Examples:

```
Jog.0.PlusInitiate              'Vel=27.2in/s
Jog.1.PlusInitiate              'Sync,Vel=1.000in/in
```

Jog.MinusInitiate

This program instruction is used to initiate jogging in the negative direction. The Jog.Stop instruction is used to stop jogging motion.

A comment is automatically inserted after the Jog.MinusInitiate instruction which shows key data about the particular jog. The comment starts with the apostrophe ‘ character.

Examples:

```
Jog.0.MinusInitiate             'Vel=27.2in/s
Jog.1.MinusInitiate             'Sync,Vel=1.000in/in
```

Jog.Stop

This program instruction is used to halt jogging using the deceleration ramp setup for the currently operating jog.

Examples:

```
  Wait For ModuleInput.2=ON       'Wait for “Jog –“ input to turn on
  Jog.0.MinusInitiate             'Vel=27.2in/s
Wait For ModuleInput.2=OFF        'Wait for “Jog –“ input to turn off
Jog.Stop                          'Decelerate to a stop

Do While(TRUE)                    'Repeat until the program is halted
   If (ModuleInput.2=ON) Then     'Jog+ when ModuleInput.2=ON
      Jog.0.PlusInitiate          'Vel=20in/s
      Wait For ModuleInput.2=OFF 'Stop when the input goes OFF
      Jog.Stop                    'Decelerate to a stop
   Endif

   If (ModuleInput.3=ON) Then     'Jog- when ModuleInput.3=ON
      Jog.0.MinusInitiate         'Vel=20in/s
      Wait For ModuleInput.3=OFF 'Stop when the input goes OFF
      Jog.Stop                    'Decelerate to a stop
   Endif
Loop
```

Dwell For Time

This motion instruction is used to pause program execution for a very precise amount of time. It operates as a motion instruction – similar to an index, home or jog. Like all other motion instructions it will not start until the preceding motion instruction has completed. A "Wait for Index.AnyCommandComplete" is not required. Likewise, any subsequent motion commands will wait and start after the dwell has completed. The total time required to complete a sequence of indexes and "Dwell For Time" instructions is extremely repeatable.

The "Dwell For Time" instruction is in units of seconds with a resolution of milliseconds (0.000 seconds).

If you want to pause the program while an index is executing you should use a "Wait for Time" instruction described below.

A comment is automatically inserted after the "Dwell For Time" instruction which notes that the dwell time is in units of seconds. The comment starts with the ' character.

Examples:

```
Do While (TRUE)
    Index.0.Initiate              'Incremetal,Dist=25.000in,Vel=25in/s
    Dwell For Time 1.000          'Seconds
Loop
```

```
Do While (TRUE)
    Index.0.Initiate              'Incremetal,Dist=25.000in,Vel=25in/s
    Dwell For Time 1.000          'Seconds
    Index.1.Initiate              'Incremental,Dist=15.000in,Vel=25in/s
    Dwell For Time 0.500          'Seconds
Loop
```

Program Flow Instructions

If/Then/Endif

This is a program flow control instruction used to selectively run a section of code only if a logical test condition is true. If the test evaluates to true the code between the If/Then and Endif lines is executed. If the test evaluates to false the code is not executed and the program skips to the next line of code after the Endif.

Logical tests (AND, OR, NOT) can be used in the If/Then/Endif instruction. Parenthesis "()" can be used to group the logical tests.

Examples:

```
If DriveInput.1=ON Then           'Turn Outputs 1 On and 2 Off if Drive
                                  'Input.1 is ON.
    DriveOutput.1=ON
```

```
    DriveOutput.2=OFF
Endif

If (DriveInput.1=ON AND DriveInput.2=OFF) Then
                                  'Turn Outputs 1 On and 2 Off if Drive
                                  'Input.1 is ON and DriveInput.2 is OFF
    DriveOutput.1=ON
    DriveOutput.2=OFF
Endif

If (ModuleInput.2=ON) Then        'Jog+ when ModuleInput.2=ON
    Jog.0.PlusInitiate            'Vel=20in/s
    Wait For ModuleInput.2=OFF    'Stop when the input goes OFF
    Jog.Stop                      'Decelerate to a stop
Endif

If (ModuleInput.3=ON) Then        'Jog- when ModuleInput.3=ON
    Jog.0.MinusInitiate           'Vel=20in/s
    Wait For ModuleInput.3=OFF    'Stop when the input goes OFF
    Jog.Stop                      'Decelerate to a stop
Endif
```

Else

This program flow instruction is used in conjunction with the If/Then/Endif instruction. If the If/Then test condition evaluates to true the code after the If/Then and before the Else is executed. If the test evaluates to false the code between the Else and the Endif is executed.

Examples:

```
If DriveInput.1=ON Then           'The following two lines are executed if
                                  'DriveInput.1=ON
   DriveOutput.1=ON
   DriveOutput.2=OFF
Else                              'The following two lines are executed if
                                  'DriveInput.1=OFF
   DriveOutput.1=OFF
   DriveOutput.2=ON
Endif

If (ModuleInput.5=ON) Then        'Set fast velocity if ModuleInput.5 = ON
   Jog.0.Vel = 1.0 'in/s
Else                              'Set slow velocity if ModuleInput.5 = OFF
   Jog.0.Vel = 0.1 'in/s
Endif
```

For Count/Next

This instruction is used to execute section of code a specific number of times.

Examples:

```
For Count = 1 to 5
    Index.1.Initiate              'Incremetal,Dist=5.250in,Vel=10.0in/s
    Dwell For Time 1.000          'seconds
Next

For Count = 1 To 10
    Wait For ModuleInput.1 = ON
    Index.0.Initiate              'Incremetal,Dist=5.000in,Vel=50in/s
    Wait For Index.AnyCommandComplete
    ModuleOutput.1=ON             'Turn ModuleOutput.1 On
    Wait For Time 1.000           'seconds
    ModuleOutput.1=OFF            'Turn ModuleOutput.1 Off
Next
```

Do While/Loop

This program instruction is used for repeating a sequence of code as long as an expression is true. To loop forever use "TRUE" as the test expression as shown in the third example below. The test expression is tested before the loop is entered. If the test expression is evaluated as False (0) the code in the loop will be skipped over.

Logical tests (AND, OR, NOT) can be used in the Do While/Loop instruction. Parenthesis "()" can be used to group the logical tests.

Examples:

```
  Do While ModuleInput.1=ON         'Repeat the three lines of code below
                                    'as long as ModuleInput.1 is ON.
     Index.1.Initiate               'Incremental,Dist=5.250in,Vel=10.0in/s
     Dwell For Time 1.000           'seconds
  Loop

  Do While (ModuleInput.1=ON AND ModuleInput.2=OFF)
                                    'Repeat the three lines of code below
                                    'as long as ModuleInput.1 is ON and
                                    'ModuleInput.2=OFF.
     Index.1.Initiate               'Incremental,Dist=5.250in,Vel=10.0in/s
     Dwell For Time 1.000           'seconds
  Loop

  Do While (TRUE)                   'Repeat until the program is halted
     Index.1.Initiate               'Incremental,Dist=5.250in,Vel=10.0in/s
```

```
    Dwell For Time 1.000          'seconds
Loop
```

Wait For

This program flow instruction is used to halt program execution until an expression becomes true. Once the expression becomes true the program continues on with the next line of code.

Logical tests (AND, OR, NOT) can be used in the Wait For instruction. However, "analog" variable comparisons (MasterPosn>1000.000) cannot be used. Only sources (DriveInput.1=ON, AtVel, etc.) can be used in a Wait For instruction.

Examples:

```
Wait For (ModuleInput.1=ON AND ModuleInput.2=OFF)
Index.0.Initiate
Wait For Index.AnyCommandComplete

If (ModuleInput.2=ON) Then        'Jog+ when ModuleInput.2=ON
    Jog.0.PlusInitiate            'Vel=20in/s
    Wait For ModuleInput.2=OFF    'Stop when the input goes OFF
    Jog.Stop                      'Decelerate to a stop
Endif

If (ModuleInput.3=ON) Then       'Jog- when ModuleInput.3=ON
   Jog.0.MinusInitiate           'Vel=20in/s
   Wait For ModuleInput.3=OFF    'Stop when the input goes OFF
   Jog.Stop                      'Decelerate to a stop
Endif
```

Wait For Time

This program instruction is used to halt program execution for a specified period of time. This instruction is not a motion instruction and can be used while a motion instruction is executing. Units: Seconds, Resolution: 0.001 seconds

A comment is automatically inserted after the "Wait For Time" instruction which notes that the time is in units of seconds. The comment starts with the apostrophe ' character.

Examples:

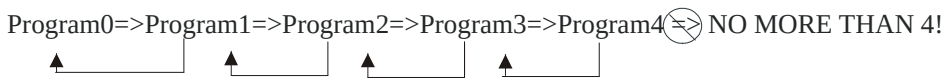
```
Wait For Time 5.000              'seconds

Do While (TRUE)                  'Repeat until the program is halted
   Index.1.Initiate              'Incremental,Dist=25.250in,Vel=10.0in/s
   Wait For AtVel
                                 'Turn Output 1 ON for 1 second, after the
                                 'index reaches its' target velocity
   DriveOutput.1=ON
```

```
   Wait For Time 1.000          'seconds
   DriveOutput.1=OFF
   Wait For Index.AnyCommandComplete
Loop
```

Call Program

This program flow instruction is used to call another program. When the called program finishes the controller picks up where it left off in the program that called it. This is often used when a section of code is used in multiple places in a program. By moving the code to another program and calling that program the total number of program lines can be reduced. This can also make the program easier to understand. Care should be taken not to "nest" more than four program calls due to processor stack overflow. Therefore, no more than four different programs should be called without returning to the original program.

In the diagram below, => Represents a Call Program instruction



Examples:

```
Call Program.10                 'Program 10 contains a complex home
                                'routine.


Call Program.100                'Program 100 contains a "teach position"
                                'routine.
```

End

This program flow instruction is used to halt the execution of the program. It can be used in multiple places within the program. It is not required on the last line of every program. It is implied that after the controller executes the last line of code in a program the program will halt. It is commonly used inside of If/Then/Endif constructs to end the program if a certain condition has been met.

Examples:

```
If DriveInput.1=OFF Then
    End
Endif

If DriveInput.1=ON Then
    DriveOutput.1=ON
    End
Endif
```

Formula

This program instruction can be used to enter a formula or assignment into a program. All FM-3 parameters are available for use in a formula. They may be dragged and dropped into a formula, but the program User Level will determine how many appear for dragging and dropping (see the section on User Level in the Setting Up Parameters chapter). Formulas can also be created by simply typing them into the program. This instruction was created to inform the user that formulas can be used in a program.

Examples:

```
Index.1.Vel = 20.0

Index.0.Dist = Index.2.Dist + 0.1

ModuleOutput.1 = ON

Index.0.Accel = (Index.0.Accel*1000)+5.00
```

Example Programs

Out and Return - Simple

Description: Move out to an absolute position and return

```
Index.2.Initiate                    'Absolute,Posn=10.000in,Vel=5.0in/s
Index.1.Initiate                    'Absolute,Posn=0.000in,Vel=10.0in/s
Wait For Index.AnyCommandComplete
```

Out and Return – More Complex

Description: Home, Wait For an input, Move out to an absolute position, set an output, dwell for 1 second, clear the output, return to home position, repeat the out and return sequence until the stop input halts the program.

```
Home.0.InitiateHome.1.Initiate      'Sensor,Offset=0.000in,Vel=-10.0in/s
ModuleOutput.1=ON                   'Set the "At Position 1" output

Do While (TRUE)                     'Repeat until the program is halted
    Wait For ModuleInput.2=ON       'Wait for the "Go" Input
    ModuleOutput.1=OFF              'Clear the "At Position 1" output
    Index.2.Initiate                'Absolute,Posn=10.000in,Vel=5.0in/s
    Wait For InPosn
    ModuleOutput.2=ON               'Set the "At Position 2" output
    Wait For Time 1.000             'Seconds
    ModuleOutput.2=OFF              'Clear the "At Position 2" output
    Index.1.Initiate                'Absolute,Posn=0.000in,Vel=10.0in/s
    Wait For InPosn
    ModuleOutput.1=ON               'Set the "At Position 1" output
Loop
```

Punch a Hole in a Web a Specified Distance Beyond a Registration Mark

Description: Index a web to a position 2 inches beyond a registration mark. Then fire a solenoid to punch a hole in the web. Wait for a sensor to indicate that the punch is in the down position. Retract the solenoid. Wait until it is sensed in the up position.

```
Do While (TRUE)                 'Repeat until the program is halted
    Index.0.Initiate            'Registration,Offet=2.0in,Dist=20in,Vel=20in/s
    Wait For InPosn
    ModuleOutput.1=ON           'Fire the punch solenoid
    Wait For ModuleInput.2=ON   'Wait for the "down" indicator
    ModuleOutput.1=OFF          'Retract the punch solenoid
    Wait For ModuleInput.3=ON   'Wait for the "up" indicator
Loop
```

Registration Index to Place a Product on a Conveyor After Each Lug

Registration Index (synchronized) to find the front edge of product, wait for input from a lug sensor and repeat.

```
Do While (TRUE)                     'Repeat until the program is halted
    Index.0.Initiate                'Sync,Registration,Offset=0.500,
                                    'Dist=50.000,Vel=1.000in/in
                                    'Registration move to product sensor.
                                    'Go to head of next product.
    Wait For Index.AnyCommandComplete
    Wait For ModuleInput.2=ON       'Wait for lug sensor on master conveyor.
Loop
```

Elevator (Accumulator) with 100 Stop Positions

Home, when an input goes on move down to the next position. When the bottom position is reached, move back to home when the input goes on.

```
Home.0.Initiate                     'Sensor,Offset=0.00mm,Vel=100mm/s

Do While (TRUE)                     'Repeat until the program is halted
    For Count = 2 To 100            'Step to positions 2 - 100
        Wait For ModuleInput.2=ON   'Wait for "Go" input
        Index.2.Initiate            'Incremetal,Dist=2.00mm,Vel=100mm/s
        Wait For InPosn
    Next

    Wait For ModuleInput.2=ON       'Wait for "Go" input
    Index.1.Initiate                'Absolute,Posn=0.00mm,Vel=1000mm/s
    Wait For InPosn
Loop
```

Simple Jogging within a Program

Jog+ when DriveInput.2 goes ON and stop when it goes off. Jog- when DriveInput.3 goes ON and stop when it goes off. This could also be accomplished using the Jog input functions when there is no program running.

```
Do While (TRUE)                     'Repeat until the program is halted
    If(DriveInput.2=ON) Then        'Jog+ when DriveInput.2=ON
        Jog.0.PlusInitiate          'Vel=20in/s
        Wait For DriveInput.2=OFF   'Stop jogging when DriveIput.2 goes OFF
        Jog.Stop                    'Decelerate to a stop
    Endif
    If (DriveInput.3=ON) Then       'Jog- when DriveInput.3=ON
        Jog.0.MinusInitiate         'Vel=20in/s
        Wait For DriveInput.3=OFF   'Stop jogging when DriveInput.3 goes OFF
        Jog.Stop                    'Decelerate to a stop
    Endif
Loop
```

Rotary Table with "Calibrated" Stop Positions

Home the axis, wait for an input and then index to 3 different stop positions (absolute positions), wait for an input between indexes. The InPosn output function could be assigned to an output to indicate when the axis has completed the index and the following error is less than a specified amount. Since the indexes are to absolute positions they can be adjusted to "calibrate" the stop positions to account for mechanical non-linearity in the particular rotary table. A rollover position of 360.00 degrees would be entered into the setup screen so that the system would take the shortest path (across the rollover) during the last move.

```
Home.0.Initiate                    'Sensor,Offset=0.0deg/s,Vel=-1000deg/s

Do While (TRUE)                    'Repeat until the program is halted
    Wait For ModuleInput.1=ON      'ModuleInput.1 is the "Go" input
    Index.2.Initiate               'Absolute,Posn=120.07deg,Vel=1000deg/s
    Wait For InPosn
    Wait For ModuleInput.1=ON
    Index.3.Initiate               'Absolute,Posn=239.95deg,Vel=1000deg/s
    Wait For InPosn
    Wait For ModuleInput.1=ON
    Index.1.Initiate               'Absolute,Posn=0.03deg,Vel=1000deg/s
    Wait For InPosn
Loop
```

Flying Cutoff/Shear

Flying cutoff or flying shear application to perform synchronized out and return indexes which repeat every 100 inches of master travel.

Part Length = 100 inches

Knife Travel Distance = 20 inches

PLS.0 is used to initiate Index.0 every 100 inches. PLS.0 has an "ON" point at 0.000 inches and an "OFF" point at 90.000 inches. PLS.0 has a rollover position of 100.000 inches. The rollover position is used to set the part length. The source for PLS.0 is the master axis. The PLS is configured in the PLS setup screen. The PLS output does not necessarily need to be connected to an output line on the drive or module because it is used within the program to initiate an index.

An Index PLS is used to fire the cutoff knife. The Index PLS is connected to Output #1 on the drive. The Index PLS is configured in the index setup screen. The Index PLS for Index.1 has an "ON" point 2.000 inches into the index and an "OFF" point 18 inches into the index.

```
Home.0.Initiate                    'Sensor,Offset=1.000in,Vel=-5.0 in/s

ModuleEncoderInput.DefineHome=ON   'Set the master position to 0.0
PLS.0.Enable=ON                    'Turn on PLS.0

Do While (TRUE)                    'Repeat until the program is halted
    Wait For DriveInput.1=ON       'Input 1 is used as a "hold" input.
    If PLS.0 = ON Then             'If the PLS is already on you are too late.
        ModuleOutput.4=ON          'Set a "Too Late" output
        End                        'Drop out of the program
    Endif
    Wait For PLS.0=ON              'Start the Index when PLS.0 goes on
                                   '(every 100 inches).
    Index.1.Initiate               'Incremental,Sync,Dist=20.0in,Vel=1.0in/in
    Index.0.Initiate               'Absolute,Sync,Posn=0.0in,Vel=2.0in/in
    Wait For Index.AnyCommandComplete
Loop
```

Synchronized Jog with Manual Phase Adjustment

The motor controls a lugged conveyor belt which is synchronized to another lugged conveyor belt. Jog.0 is configured as a "Synchronized" jog using the setup software. The program first homes the follower and then waits for an input from a sensor on the master axis lugs. When the input comes on the follower starts the synchronized jog. If the home is setup correctly the follower will be in perfect phase when it gets up to speed. If the follower gets out of phase with the master the operator can manually bring the it back into phase using "Advance" and "Retard" inputs. The program adjusts the phase of the follower axis by adjusting the jog velocity (Jog.0.Vel) when the operator hits one of the phasing inputs.

```
Home.0.Initiate                     'Sensor,Offset=2.25in,Vel=10in/s
Jog.0.Vel=1.000                     'follower inches/master inch

Wait For ModuleInput.1=ON           'Start when a master lug is detected
Jog.0.PlusInitiate                  'Sync,Vel=1.000in/in

Do While (TRUE)                     'Repeat until the program is halted
    If (ModuleInput.2=ON) Then      'Phase Advance when ModuleInput.2=ON
        Jog.0.Vel=1.100             'follower inches/master inch
        Wait For ModuleInput.2=OFF
        Jog.0.Vel=1.000             'follower inches/master inch
    Endif

    If (ModuleInput.3=ON) Then      'Phase Retard when ModuleIput.3=ON
        Jog.0.Vel=0.900             'follower inches/master inch
        Wait For ModuleInput.3=OFF
        Jog.0.Vel=1.000             'follower inches/master inch
    Endif

Loop
```

Auger Filler with Inputs to Adjust the Fill Amount

Incremental indexes are used to squirt a specified amount of food product into a box. Inputs are used to adjust the index distance. It would be much simpler to adjust the index distance with an OIT-3165 operator interface panel, but inputs could be used as described below.

```
ModuleOutput.3=OFF
ModuleOutput.4=OFF

Do While (TRUE)                     ‘Repeat until the program is halted
    If (ModuleInput.2=ON) Then      ‘Fill a box if the “Go” input is on.
        Index.1.Initiate            ‘Incremental,Dist=16.00oz,Vel=16.0oz/s
        Wait For Index.AnyCommandComplete
    Endif

‘Increase the fill amount once every time ModuleInput.3 is pressed
    If((ModuleInput.3=ON) AND (ModuleOutput.3=OFF)) Then
        Index.1.Dist = Index.1.Dist + 0.10    ‘ounces
        ModuleOutput.3=ON

‘ModuleOutput.3 is used to make sure that the distance is
‘incriminated only once each time ModuleInput.3 is pressed.
        If (Index.1.Dist > 20) Then
            ModuleOutput.1=ON       ‘Long index output
        Else
            ModuleOutput.1=OFF
        Endif
    Endif

    If((ModuleInput.3=OFF) AND (ModuleOutput.3=ON)) Then
        ModuleOutput.3=OFF
    Endif

‘Increase the fill amount once every time ModuleInput.4 is pressed.
    If ((ModuleInput.4=ON) AND (ModuleOutput.4=OFF)) Then
        Index.1.Dist = Index.1.Dist - 0.10 ‘ounces
        ModuleOutput.4=ON

‘ModuleOutput.4 is used to make sure that the distance is
‘incriminated only once each time ModuleInput.4 is pressed.
        If (Index.1.Dist < 12) Then
            ModuleOutput.2=ON       ‘Short index output
        Else
            ModuleOutput.2=OFF
        Endif
    Endif

    If((ModuleInput.4=OFF) AND (ModuleOutput.4=ON)) Then
        ModuleOutput.4=OFF
    Endif
Loop
```

Sequence Learn and Playback

This example consists of three programs. The first program is used to learn 3 positions using “Jog+”, “Jog-“, “Jog Fast” and “Learn” inputs. The second program is called several times by the first program. The third program steps through the learned positions

Learn Program (Program 0)

```
Home.0.Initiate                     ‘Sensor,Offset=0.000in,Vel=-10in/s

Index.1.Initiate                    ‘Move to position 1
Wait For Index.AnyCommandComplete

Call Program.1                      ‘Program 1 allows the axis to be jogged
                                    ‘into position
If (ModuleInput.1=ON) Then          ‘Learn the new position if the “Learn”
                                    ‘input is on
    Index.1.Dist = PosnCommand      ‘Read the Position Command into Index.1’s
                                    ‘absolute position.
Endif
Wait For ModuleInput.1=OFF          ‘Wait until the “Learn” input goes off
Wait For ModuleInput.2=OFF          ‘Wait until the “Skip” input goes off

Index.2.Initiate                    ‘Move to position 2
Wait For Index.AnyCommandComplete

Call Program.1                      ‘Program 1 allows the axis to be jogged
                                    ‘into position
If (ModuleInput.1=ON) Then          ‘Learn the new position if the “Learn”
                                    ‘input is on
    Index.2.Dist = PosnCommand      ‘Read the Position Command into Index.2’s
                                    ‘absolute position.
Endif
Wait For ModuleInput.1=OFF          ‘Wait until the “Learn” input goes off
Wait For ModuleInput.2=OFF          ‘Wait until the “Skip” input goes off

Index.3.Initiate                    ‘Move to position 3
Wait For Index.AnyCommandComplete

Call Program.1                      ‘Program 1 allows the axis to be jogged
                                    ‘into position
If (ModuleInput.1=ON) Then          ‘Learn the new position if the “Learn”
                                    ‘input is on
    Index.3.Dist = PosnCommand      ‘Read the Position Command into Index.3’s
                                    ‘absolute position.
Endif
Wait For ModuleInput.1=OFF          ‘Wait until the “Learn” input goes off
Wait For ModuleInput.2=OFF          ‘Wait until the “Skip” input goes off
```

Subroutine for Jogging the Axis into the Desired Position (Program 1)

```
'Allow jogging until either the "Learn" input (ModuleInput.1)
'or the "Skip" input (ModuleInput.2) goes ON.

Do While ((ModuleInput.1=OFF) AND (ModuleInput.2=OFF))
    If (ModuleInput.3=ON) Then              'Jog+ if the Jog+ input is on
        Jog.0.PlusInitiate                  'Vel=0.1in/s

        Do While (ModuleInput.3=ON)
            If (ModuleInput.5=ON) Then      'ModuleInput.5 = "Jog Fast"
                Jog.0.Vel = 1.0             'in/s
            Else
                Jog.0.Vel = 0.1             'in/s
            Endif
        Loop

        Jog.Stop                    'Stop jogging when the Jog+ input goes off.
    Endif

    If (ModuleInput.4=ON) Then              'Jog- if the Jog- input is on
        Jog.0.MinusInitiate                 'Vel=0.1in/s

        Do While (ModuleInput.4=ON)
            If (ModuleInput.5=ON) Then      'ModuleInput.5 = "Jog Fast"
                Jog.0.Vel = 1.0             'in/s
            Else
                Jog.0.Vel = 0.1             'in/s
            Endif
        Loop

        Jog.Stop                    'Stop jogging when the Jog+ input goes off.
    Endif

Loop
```

Playback Program (Program 2)

```
Home.0.Initiate                    'Sensor,Offset=0.000in,Vel=-10in/s

Do While (TRUE)                    'Repeat until the program is halted
    Index.1.Initiate               'Absolute,Posn=1.000in,Vel=5in/s
    Wait For InPosn
    ModuleOutput.1=ON              'Turn on ModuleOutput.1 for 1 second
    Wait For Time 1.000            'seconds
    ModuleOutput.1=OFF

    Index.2.Initiate               'Absolute,Posn=20.000in,Vel=7in/s
    Wait For InPosn
    ModuleOutput.1=ON              'Turn on ModuleOutput.1 for 1 second
    Wait For Time 1.000            'seconds
    ModuleOutput.1=OFF

    Index.3.Initiate               'Absolute,Posn=5.250in,Vel=10in/s
    Wait For InPosn
    ModuleOutput.1=ON              'Turn on ModuleOutput.1 for 1 second
    Wait For Time 1.000            'seconds
    ModuleOutput.1=OFF
Loop
```

Parameter Descriptions

This section lists all programmable and feedback parameters available. The parameters are listed alphabetically by variable name on the right and give a description. Range is dynamic and depends on User Unit scaling. The units of the parameters are dynamic and depend on selected User Units.

Absolute Position Valid — **AbsolutePositionValid**

This source is activated when either the DefineHome destination is activated, or any home is successfully completed (sensor or marker found). This source is deactivated if the drive is rebooted, an encoder fault occurs, the drive is powered down, or a home is re-initiated.

Accelerating — **Accelerating**

This source is active when the FM-3 is executing an acceleration ramp. A normal index consists of 3 segments: Accelerating, At Velocity, and Decelerating. The Accelerating source will be set (active) during this acceleration segment regardless of whether the motor is speeding up or slowing down. Therefore, this source can sometimes be active when the motor is decelerating. This could be true when compounding indexes together.

Acceleration Type — **AccelType**

This parameter is used to select the accel/decel type for all motion (homes, jogs and indexes). The "S-Curve" ramps offer the smoothest motion, but lead to higher peak accel/decel rates. "Linear" ramps have the lowest peak accel/decel rates but they are the least smooth ramp type. "5/8 S-Curve" ramps and "1/4 S-Curve" ramps use smoothing at the beginning and end of the ramp but have constant (linear) accel rates in the middle of their profiles. The "5/8 S-Curve" is less smooth than the "S-Curve" but smoother than the "1/4 S-Curve". S-Curve accelerations are very useful on machines where product slip is a problem. They are also useful when smooth machine operation is critical. Linear ramps are useful in applications where low peak torque is critical. Below is a comparison of the 4 ramp types:

- S-Curve: Peak Accel = 2 x Average Accel
- 5/8 S-Curve: Peak Accel = 1.4545 x Average Accel
- 1/4 S-Curve: Peak Accel = 1.142857 x Average Accel
- Linear: Peak Accel = Average Accel

Acceleration Decimal Places — **AccelUnits.Decimal**

This parameter is the decimal point location for all real-time accel./decel. ramps.

Acceleration Time Scale	AccelUnits.TimeScale

This parameter is the time units for accel./decel. ramps. Possible selections are milliseconds or seconds.

At Velocity	AtVel

This source is active when the FM-3 is executing a constant velocity motion segment. One example would be during an index. The source would activate after the motor has finished accelerating up to speed and before the motor begins to decelerate to a stop. A normal index consists of 3 segments: Accelerating, At Velocity, and Decelerating. This source is active during the At Velocity segment, and is activated based on the commanded velocity, not the feedback velocity. During synchronized motion, AtVel can be active without actual motor movement.

Brake Activate	Brake.Activate

This destination, when activated, engages the brake. This is simply used to manually engage the brake outside of the normal brake operation. This is level sensitive.

Brake Disengaged	Brake.Disengaged

This source is used to control the motor holding brake. When it is "off" the brake is mechanically engaged. When the brake is engaged, the diagnostic display on the front of the drive will display a "b". The drive and module outputs are limited to 150 mA capacity, therefore, a suppressed relay is required to control the brake coil. Model BRM-1 may be used.

Brake Release	Brake.Release

This destination will release the brake under all conditions, even when Brake.Activate is engaged. When this input function is active, the Brake.Disengaged output function (source) will be activated. This is used as a manual brake override.This is level sensitive.

Commanding Motion	CommandingMotion

This source activates when VelCommand is non-zero.

Decelerating

This source is active when the FM-3 is decelerating. A normal index consists of 3 segments: Accelerating, At Velocity, and Decelerating. Decelerating follows the accelerating segment and the At Velocity segment. When indexes are compounded to create a complex motion profile, only the last index may contain a decelerating segment.

Define Home **DefineHome**

This destination is used to set the Commanded Position to the value specified in the DefineHomePosn variable. On the rising edge of this input function the absolute position is set equal to the DefineHomePosn and the AbsolutePosnValid output function (source) is activated.

Define Home Position **DefineHomePosn**

The DefineHome parameter is used to set the motors absolute position to the value stored in the DefineHomePosn variable. On the rising edge of the DefineHome function the Commanded Position is set equal to the DefineHomePosn and the AbsolutePosnValid source is activated.

Distance Characteristic Distance **DistUnits.CharacteristicDist**

This parameter is the distance the load travels (in user units) when the motor travels the characteristic length (in motor revolutions). This parameter is used along with the DistUnits.CharacteristicLength to establish the relationship between user distance and actual motor travel distance. See the section on the User Units View in the Setting Up Parameters chapter.

Distance Characteristic Length **DistUnits.CharacteristicLength**

This parameter is the distance the motor travels (in whole number of revolutions) to achieve one characteristic distance of load travel. This parameter is used along with the DistUnits.CharacteristicDist to establish the relationship between user distance and motor travel distance. See the section on the User Units View in the Setting Up Parameters chapter.

Distance Decimal Places **DistUnits.Decimal**

This parameter is used to select the number of decimal places used in the DistUnits.CharacteristicDist. Using a high number of decimal places will improve positioning resolution, but will also limit the maximum travel distance. The number of decimal places set in this parameter determines the number of decimal places used in all distance parameters throughout the software. You can select from zero to six decimal places of accuracy.

Distance Name **DistUnits.Name**

This is a text variable which is used as the label for the distance/position user units. It can be up to 12 characters in length.

Drive Ambient Temperature **DriveAmbientTemp**

This parameter should be set to reflect the ambient air temperature near the drive heatsink during normal operation. This will determine the amount of regenerative power that can be dissipated by the E Series drive's internal shunt resistor. When that calculated value is exceeded by the Shunt Power RMS parameter a shunt fault will occur. Valid only for EN-208 and EN-214 drives.

Drive Analog Output Feedback **DriveAnalogOutput.#.Feedback**

Displays the Output voltage from one of the two analog outputs found on the 3-pin connector on the front of the drive or on the drive command connector.

Drive Enable Status **DriveEnableStatus**

This source is active when the drive is enabled.

Drive Input Debounced **DriveInput.#.Debounced**

This displays the state of the input after the debounce is taken into account.

Drive Input Debounce Time **DriveInput.#.DebounceTime**

The Drive Input Debounce Time parameter is the minimum time a digital input must be steady in order to be recognized by the FM-3. This feature helps prevent false triggering in applications in electrically noisy environments.

Drive Input Force **DriveInput.#.Force**

Input can be forced either On or Off. This parameter is the state to which the input will be forced when the ForceEnable bit is activated.

Drive Input Force Enable **DriveInput.#.ForceEnable**

If DriveInput.#.ForceEnable parameter is activated, then the state of the DriveInput.#.Force bit will override the current input state.

Drive Input Name **DriveInput.#.Name**

This is a text string up to 12 characters that can be assigned to a given input. It allows the user to use application specific terminology in naming digital inputs.

Drive Input Raw **DriveInput.#.Raw**

This displays the raw state of the digital input without debounce or forcing to override the raw status.

Drive Input Status **DriveInput.#.Status**

This source is the state of the input after debounce and forcing are taken into account.

Drive Output Force **DriveOutput.#.Force**

A drive output can be forced either On or Off with this parameter. If the ForceEnable bit is activated, the DriveOutput.#.State will be set to this value.

Drive Output Force Enable **DriveOutput.#.ForceEnable**

If DriveOutput.#.ForceEnable parameter is activated, then the state of the DriveOutput.#.Force bit will override the current output state.

Drive Output Name **DriveOutput.#.Name**

This is a text string up to 12 characters that can be assigned to a given output. It allows the user to use application specific terminology in naming digital outputs.

Drive Output State **DriveOutput.#State**

This destination sets the current state of an output line.

Drive Output Encoder Scaling Enable **DriveOutputEncoder.ScalingEnable**

When on, this parameter enables the use of the drive encoder output scaling feature.

Drive Output Encoder Scaling **DriveOutputEncoder.Scaling**

This parameter allows scaling of the drive encoder output resolution in increments of one line per revolution. Allowable range is from one line per revolution up to the actual density of the encoder in the motor. If the Encoder output scaling is set greater than the motor encoder density the output scaling will be equal to the motor encoder density.

Drive Serial Number **DriveSerialNumber**

This displays the serial number of the Drive to which the FM-3 is attached.

Drive Faults Bitmap	Fault.DriveFaultsBitmap

This parameter is a 32-bit register which holds all of the drive fault status bits. Following is a list of all drive faults and their associated bit numbers:

0 = Encoder state fault

1 = Encoder hardware fault

3 = Drive power module fault

4 = Low DC bus fault

5 = High DC bus fault

8 = Drive internal fault 1

9 = Drive trajectory fault

10 = Drive internal fault 2

16 = Drive watchdog timer fault

19 = Drive over speed fault

21 = Drive power up self test fault

20 = Drive invalid configuration fault

24 = Drive RMS shunt power fault

25 = Motor overtemperature fault

All other bits are not used. A "1" in these bit locations indicates the specific fault is active, and a "0" is inactive.

Drive OK	Fault.DriveOK

Active when there are no faults. Inactivated when any fault except travel limits occur. Drive enable has no effect on this event.

Faulted	Fault.Faulted

Any fault will activate this event.

Active Fault	Fault.#.Active

The specified fault is active. See the help index for more information on faults and recovery from them.

Fault Counts	Fault.#.Counts

The module stores the total number of times the specific fault has occurred since it was manufactured.

Module Faults Bitmap	Fault.ModuleFaultsBitmap

This parameter is a 32-bit register which holds all of the module fault status bits. Following is a list of all module faults and their associated bit numbers:

0 = Module watchdog timer fault

1 = Module invalid configuration fault

2 = Module NVM invalid fault

3 = Module power up self test fault

4 = Module following error fault

5 = Module travel limit plus

6 = Module travel limit minus

7 = Module program fault

All other bits are not used. A "1" in these bit locations indicates the specific fault is active, and a "0" is inactive.

Reset Faults	Fault.Reset

Resets faults that do not require a power down. This event is "or"ed with the reset button on the drive.

Enable Feedforwards	FeedforwardsEnable

This parameter may be setup on the Tuning view or through a program, and enables feedforward compensation. When feedforwards are enabled, the accuracy of the Inertia and Friction settings are very important. If the Inertia setting is larger than the actual inertia, the result could be a significant overshoot during ramping. If the Inertia setting is smaller than the actual inertia, following error during ramping will be reduced but not eliminated. If the Friction is greater than the actual friction, it may result in velocity error or instability. If the Friction setting is less than the actual friction, velocity error will be reduced, but not eliminated.

Feedhold	Feedhold

When this destination is activated the motor will decelerate to a stop in the time specified by the FeedholdDecelTime parameter. When it is deactivated the motor will accelerate back up

to the programmed speed in the same amount of time. It is used to hold motion without cancelling the move in progress. If a feedhold is activated during an index the motor will come to a halt, but the index's velocity command remains at the velocity it was at before the feedhold was activated. When the feedhold is deactivated time will ramp back up and the index will continue on to its programmed distance or position. Feedhold affects indexes, homes, and programs. A jog is not affected by the feedhold unless it is initiated from a program. This is level sensitive.

Feedhold Deceleration Time **FeedholdDecelTime**

When Feedhold destination is activated the motor will decelerate to a stop in the time specified by the FeedholdDecelTime parameter. While the feedhold destination is active, the motion profile is stopped.

FeedRate Override **FeedRateOverride**

This parameter is used to scale all motion. It can be described as “scaling in real time.” The default setting of 100% will allow all motion to occur in real time. A setting of 50% will scale time so that all moves run half as fast as they do at 100%. A setting of 200% will scale time so that all moves run twice as fast as they would at 100%. Feed Rate Override is always active, and this parameter may be modified via Modbus or in a program. When changed, the new value takes effect immediately.

Foldback Active **FoldbackActive**

This source (output function) is active when the drive is limiting motor current. If the Foldback RMSexceeds 100 percent of the continuous rating, the current foldback circuit will limit the current delivered to the motor to 80 percent of the continuous rating.

Foldback RMS **FoldbackRMS**

This read-only parameter accurately models the thermal heating and cooling of the drive. When this parameter reaches 100 percent, current foldback will be activated.

Following Error **FollowingError**

Following Error displays the difference between the Position Command and the Position Feedback.

Enable Following Error **FollowingErrorEnable**

This parameter can be setup from the Position view or from a program. When enabled, a following error fault will be generated if the absolute value of the Following Error exceeds the Following Error Limit.

Following Error Limit **FollowingErrorLimit**

This parameter is used when the FollowingErrorEnable bit is set. This limit is compared to the absolute value of the FollowingError. If the FollowingError is greater than the FollowingErrorLimit, a following error fault will be generated.

Friction

This parameter is characterized in terms of the rate of friction increase per 100 motor RPM. If estimated, always use a conservative (less than or equal to actual) estimate. If the friction is completely unknown, a value of zero should be used. A typical value used here is less than one percent.

Any Command Complete **Home.AnyCommandComplete**

This source is active when any home motion command is completed, if a stop is activated before the home has completed the function will not be activated. Inactivated when a home command is executed.

Accelerating **Home.#.Accelerating**

Active during any acceleration while the specified home is in progress. Accelerating may turn off and on again based on the type of Home selected. Accelerating will activate during the Home back off sensor motion.

Acceleration **Home.#.Accel**

This parameter sets the average Acceleration rate used during the home, units are specified on the User Units page.

At Velocity **Home.#.AtVel**

This source is activated when the home velocity is reached when a the specified home is in progress. It will activate and deactivate base on the home. Home At Velocity will not be activated during back off sensor portion of the home.

If On Sensor **Home.#.OnSensorAction**

If the home sensor input is active when the home is initiated, this parameter determines the direction of motion. Two selections are possible. If "Back off before homing" is selected, the motor will turn in the opposite direction of the home until the home sensor is clear and then begin the home. If "Go forward to next sensor" is selected, the motor will turn in the commanded direction until the next rising edge of the sensor is seen. If using Modbus to view or modify this parameter, 1= Back off before homing, 0 = Go forward to next sensor.

Calculated Offset **Home.#.CalculatedOffset**

The Calculated offset is the distance travelled during the deceleration ramp from the home velocity to a stop. Calculated by Emerson Motion Control PowerTools®.

Command Complete **Home.#.CommandComplete**

This source is active when the specified home command is completed, if a stop is activated before the home has completed the function or if the Home Limit Distance has been exceeded it will not be activated. Inactive when a home command is executed.

Command In Progress **Home.#.CommandInProgress**

This source is activated when the Home is initiated and remains active until all motion related to the Home has completed.

Decelerating **Home.#.Decelerating**

This source is active during any deceleration while the specified home is in progress. Decelerating will turn off and on based on the type of Home selected. Decelerating will activate during the Home back off sensor motion.

Deceleration **Home.#.Decel**

The Deceleration ramp parameter is used during all the home moves specified in user units.

Initiate **Home.#.Initiate**

When activated, this destination will initiate the specified home. Home will not initiate if an index, jog, program, or stop is currently active.

Limit Distance **Home.#.LimitDist**

This parameter places an upper limit on the incremental distance traveled during a home. If no home reference is found the motor will decelerate to a stop at the limit distance and activate the Home.#.LimitDistHit event.

Enable Limit Distance **Home.#.LimitDistEnable**

This parameter enables the specified Home.#.LimitDist. If not enabled, the home will run indefinitely until the home reference is found.

Limit Distance Hit	Home.#.LimitDistHit

This source is activated when the home sensor is not found before the Home Limit Distance is traveled.

Name	Home.#.Name

User name for the specified home.

Home Offset Type	Home.#.OffsetType

Selects calculated or specified home offset. Calculated offset is the distance traveled during the deceleration ramp from the home velocity. The specified offset allows the user to choose an exact offset from the Home Reference.

End of Home Position	Home.#.EndPosn

This parameter defines the drive position at the completion of a home. Typically used to define the machine coordinate home position.

Home Reference	Home.#.Reference

This parameter determines how the reference position is determined. The parameter can have one of three different values: 'Sensor', 'Marker', 'Sensor then Marker'. When the home reference is 'Sensor' the rising edge of the 'Home Sensor' input function is used to establish the reference position. When the home reference is 'Marker' the rising edge of the motor encoder's marker channel is used to establish the reference position. When the home reference is 'Sensor then Marker' the reference position is established using the first marker rising edge after the Home Sensor input function goes active.

Sensor Trigger	Home.#.SensorTrigger

This destination is usually a sensor input used as a reference for the home. This event is only used if the home is defined by sensor or by sensor and marker.

Specified Offset	Home.#.SpecifiedOffset

The specified offset parameter allows the user to specify an exact offset relative to the Home Reference. The commanded motion will stop at exactly the offset distance away from the sensor or the marker as specified.

Time Base	Home.#.TimeBase

The time base selects either realtime, which allows velocities, acceleration and deceleration to be based on real time, or synchronized, which allows for an external synchronization signal.

Velocity	Home.#.Vel

This parameter sets the target velocity for all of moves in the home. The sign determines the home direction. Positive numbers cause motion in the positive direction and negative numbers cause motion in the negative direction in search of the home sensor.

Any Command Complete	Index.AnyCommandComplete

This source is active when any index motion command is completed. If a stop is activated before the index has completed, this destination will not activate. Deactivated when any index command is initiated.

Accelerating	Index.#.Accelerating

This source is active while an index is accelerating to its target velocity. Once the index reaches the target velocity, or begins to decelerate, the Index.#.Accelerating source will deactivate.

Acceleration	Index.#.Accel

This parameter is the average Acceleration rate used during the index. Units are specified on the User Units view in the PowerTools FM-3 software.

Analog	Index.#.AnalogLimitType

Select the analog parameter (i.e. Torque Command) to compare to the AnalogLimitValue. Satisfying the comparison triggers the index registration event.

Limit Value	Index.#.AnalogLimitValue

This preset value is compared to the selected AnalogLimitType to determine the registration point.

At Velocity	Index.#.AtVel

This source activates when the target index velocity is reached. If Feedrate override is changed or FeedHold is activated AtVelocity shall remain active. Index.#.AtVel will

deactivate at the start of any deceleration or acceleration. During a synchronized index, this source could be active even without any motor motion if the master axis stops.

Command Complete **Index.#.CommandComplete**

The Index.#.CommandComplete source will activate when the specific index completes its deceleration ramp. It will remain active until the specific index is initiated again. If the Stop destination is used during an Index, then the Index.#.CommandComplete will not activate.

Command In Progress **Index.#.CommandInProgress**

The Index.#.CommandInProgress source is active throughout an entire index profile. The source activates at the beginning of the index acceleration ramp, and deactivates at the end of the index deceleration ramp.

Decelerating **Index.#.Decelerating**

This source is active while an index is decelerating from its target velocity. Once the index reaches zero velocity, or its next target velocity, the Index.#.Decelerating source will deactivate.

Deceleration **Index.#.Decel**

This parameter is the Average Deceleration rate used during the index. Units are specified on the User Units page.

Distance **Index.#.Dist**

This parameter is the Incremental distance that the index will travel or the absolute position for the specified index in user units. If an index type of Registration is selected, then this is a limit distance, or the maximum distance the index will travel if a registration sensor is not seen.

Initiate **Index.#.Initiate**

The Index.#.Initiate destination is used to initiate the specific index. The Index is initiated on the rising edge of this function. An Index cannot be initiated if there is an Home, Jog, or Program in progress, or if the Stop destination or if a travel limit is active. It can be activated from an assignment or from a program.

CompoundInitiate **Index.#.CompoundInitiate**

When activated will initiate the specified Index to compound into the next index in the program. Only allowed in a program.

Limit Distance Hit **Index.#.LimitDistHit**

This source is activated when the registration sensor is not found before the Limit Distance is traveled. If the Registration Window is enabled the sensor must be activated inside the window to be recognized.

Name **Index.#.Name**

The user can specify an Index name of up to 12 characters.

Enable PLS **Index.#.PLSEnable**

When Activated, this parameter enables the PLS (programmable limit switch) function for the specified index. It can be controlled from index view check box or from a program.

PLS Off Point **Index.#.PLSOffDist**

This an incremental distance from the start of the index to the Index PLS off point. This is an unsigned value and is relative only to starting position of this index. Index direction does not affect this parameter. Index.#.PLSStatus will be active if the distance traveled from the start of the index is greater than the Index.#.PLSOnDist and less than the Index.#.PLSOffDist.

PLS On Point **Index.#.PLSOnDist**

This an incremental distance from the start of the index to the Index PLS On Point. This is an unsigned value and is relative only to starting position of this index. Index direction does not affect this parameter. Index.#.PLSStatus will be active if the distance traveled from the start of the index is greater than the Index.#.PLSOnDist and less than the Index.#.PLSOffDist.

PLS Status **Index.#.PLSStatus**

Controlled by the PLSOn and PLSOff Points, this is relative to the distance commanded since the start of the index. Index.#.PLSStatus will be active if the distance traveled from the start of the index is greater than the Index.#.PLSOnDist and less than the Index.#.PLSOffDist.

RegistrationOffset **Index.#.RegistrationOffset**

This parameter is the the Distance the motor will travel after a valid registration sensor or analog limit value has been detected.

Registration Type **Index.#.RegistrationType**

This selects either sensor or analog as the registration mark for a registration index.

Enable Registration Window **Index.#.RegistrationWindowEnable**

This check box enables (if checked) the Registration Sensor valid Window. When active, only registration marks that occur inside the registration window are seen as valid.

Window End **Index.#.RegistrationWindowEnd**

This parameter defines the end of the Registration Sensor Valid Window relative to start position of this index. This is an unsigned value and is relative only to starting position of this index. Index direction does not affect this parameter. The Registration Window start position is greater than or equal to the Registration point and less than the Registration Window End position. If a registration sensor is seen outside of this window (not between the WindowStart and WindowEnd positions) then it will be ignored.

Window Start **Index.#.RegistrationWindowStart**

This parameter defines the start of the Registration Sensor Valid Window relative to start position of this index. This is an unsigned value and is relative only to starting position of this index. Index direction does not affect this parameter. The Registration Window start position is greater than or equal to the Registration point and less than the Registration Window End position. If a registration sensor is seen outside of this window (not between the WindowStart and WindowEnd positions) then it will be ignored.

Registration Sensor **Index.#.SensorTrigger**

If registration to Sensor is selected, when this destination activates, motor position is captured and is used as the registration point for registration type indexes.

Time Base **Index.#.TimeBase**

The time base selects either realtime, which allows velocities, acceleration and deceleration to be based on real time, or synchronized, which allows for an external synchronization signal.

Velocity **Index.#.Vel**

This parameter sets the target velocity of the specific index. The units for this parameter are specified in the User Units Setup view. When an index is initiated, it will ramp up to this velocity at the specified acceleration rate and run at speed until it decelerates to a stop (assuming the index is not compounded).

Inertia Ratio	InertiaRatio

This specifies the load to motor inertia ratio. For example, a value of 25.0 specifies that the load inertia is 25 times the inertia of the motor.

Initially Active	InitiallyActive

This source, when assigned to a destination, will activate the destination on power-up or upon FM-3 reset. InitiallyActive can be assigned to any destination that does not create motion (i.e. indexes, jogs, homes, programs).

In Position	InPosn

This source activates when commanded velocity is zero and the absolute value of the following error is less than the InPosnWindow for at least the amount of time specified in the InPosnTime parameter.

In Position Time	InPosnTime

This parameter is the minimum amount of time that commanded motion must be complete and the absolute value of the following error is less than the InPosnWindow parameter for the InPosn source to activate.

In Position Window	InPosnWindow

The absolute value of the following error must be less than this value at the completion of a move for the InPosnTime before InPosn will activate.

Any Command Complete	Jog.AnyCommandComplete

The Jog.AnyCommandComplete bit will activate when either Jog 0 or Jog 1 completes its deceleration ramp and reaches zero commanded speed. It deactivates when another jog is initiated.

Accelerating	Jog.#.Accelerating

This source is active while a jog is accelerating to its target velocity. Once the jog reaches the target velocity, the Jog.#.Accererating bit will turn off.

Acceleration	Jog.#.Accel

This parameter is the average acceleration ramp for the specific jog.

At Velocity **Jog.#.AtVel**

This source activates when the particular jog has reached its target velocity. It deactivates when accelerating or decelerating to another target jog velocity.

Command Complete **Jog.#.CommandComplete**

The Jog.#.CommandComplete source activates when the specific Jog completes its deceleration ramp and reaches zero commanded speed. It deactivates when the specific Jog is initiated again.

Command In Progress **Jog.#.CommandInProgress**

The Jog.#.CommandInProgress source is high throughout an entire jog profile. The bit goes high at the start of a jog acceleration ramp, and turns off at the end of a jog deceleration ramp.

Decelerating **Jog.#.Decelerating**

This source turns on at the beginning of a jog deceleration ramp and turns off at the completion of the ramp.

Deceleration **Jog.#.Decel**

This parameter is the average deceleration ramp for the specific jog.

Initiate Minus **Jog.#.MinusInitiate**

This is used inside a program to initiate a specific jog. When this bit is active, jogging motion will be initiated in the negative direction at the specified jog velocity.

Initiate Plus **Jog.#.PlusInitiate**

This is used inside a program to initiate a specific jog. When this bit is active, jogging motion will be initiated in the positive direction at the specified jog velocity.

Time Base **Jog.#.TimeBase**

The time base selects either realtime, which allows velocities, acceleration and deceleration to be based on real time, or synchronized, which allows for an external synchronization signal.

Velocity	Jog.#.Vel

This parameter specifies the velocity used for jogging with the Jog.PlusActivate and Jog.MinusActivate destinations or the Jog.#.PlusInitiate and Jog.#.MinusInitiate inside a program. The units for this parameter are specified in the User Units view.

Select	Jog.Select0

This destination is used to select between the jogs. It is used along with the Jog.PlusActivate and Jog.MinusActivate destinations. If the Jog.Select0 destination is not active then the Jog.0 setup parameters will be used for jogging. If the Jog.Select0 input function is active, the Jog.1 setup parameters will be used for jogging. If the Jog.Select destination is changed during jogging motion the axis will ramp smoothly from the previously selected jog velocity to the new jog velocity using the specified jog acceleration. This is level sensitive.

Minus Activate	Jog.MinusActivate

This destination is used to initiate jogging motion in the negative direction using the jog parameters of the jog selected by the Jog select input function. Jogging will continue as long as the destination is active. The motor will decelerate to a stop when the destination is deactivated. This is level sensitive.

Plus Activate	Jog.PlusActivate

This destination is used to initiate jogging motion in the positive direction using the jog parameters of the jog selected by the Jog select input function. Jogging will continue as long as the destination is active. The motor will decelerate to a stop when the destination is deactivated. This is level sensitive.

Stop	Jog.Stop

This is used only in programs to halt jogging motion. Jogging motion is initiated in programs using the Jog.#.MinusActivate or Jog..#.PlusActivate instructions, and using the Jog.Stop will cause the motor to decelerate to a stop at the Jog.#.Decel rate for the jog that is active.

Line Voltage	LineVoltage

This parameter is used to calculate critical internal gains. There are two possible value: 115 or 230 VAC. A value of 115 VAC should not be used if the actual line voltage is 230, otherwise damage to the E Series drive may result. The default value is 230 VAC.

BaudRate	Modbus.BaudRate

Modbus baudrate for this FM module and drive.

Modbus Id	Modbus.ModbusId

Modbus ID # for this FM module and drive.

Debounced	ModuleInput.#.Debounced

This is the state of the input after the debounce is taken into account.

Module Input Debounce Time	ModuleInput.#.DebounceTime

The Module Input Debounce Time parameter is the minimum time a digital input must be on in order to be recognized by the FM-3. This feature helps prevent false triggering in applications in electrically noisy environments.

Module Input Force	ModuleInput.#.Force

Input can be forced either On or Off. This bit is the state to which the input will be forced when the ForceEnable bit is activated.

Module Input Enable Force	ModuleInput.#.ForceEnable

If ModuleInput.#.ForceEnable is True (or On), then the state of the ModuleInput.#.Force bit will override the current input state.

Module Input Name	ModuleInput.#.Name

This is a text string up to ten characters that can be assigned to a given input. It allows the user to use application specific terminology in naming digital inputs.

Module Input Raw	ModuleInput.#.Raw

This is the raw state of the digital input without debounce or forcing to override the raw status.

Module Input Status	ModuleInput.#.Status

This source is the state of the input after debounce and forcing are taken into account.

Module Output Force	ModuleOutput.#.Force

A module output can be forced either On or Off. If the ForceEnable bit is activated, the ModuleOutput.#.State will be set to this value.

ModuleOutput Enable Force **ModuleOutput.#.ForceEnable**

If ModuleOutput.#.ForceEnable is activated, then the state of the ModuleOutput.#.Force bit will override the current output state.

ModuleOutput Name **ModuleOutput.#.Name**

User assigned name to the hardware output.

ModuleOutput State **ModuleOutput.#State**

This destination sets the current state of an output line.

Module Serial Number **ModuleSerialNumber**

This is the FM-3 module serial number.

Motor Type **MotorType**

This parameter is used to select the motor type.

Name

User name for this FM-3 axis can have a length up to 12 characters. This can be used to help differentiate setup files.

PLS Direction **PLS.#.Direction**

This parameter specifies the direction of motion that a particular PLS output will function. If set to Both, the PLS will activate regardless of whether the axis is moving in the positive or negative direction. If set to Plus, the PLS will activate only when the axis is moving in the positive direction. If set to Minus, the PLS will activate only when the axis is moving in the negative direction. A flying cutoff or flying shear application may use this feature to activate the PLS to fire the knife only when the axis is moving in the positive direction.

PLS Off Point **PLS.#.OffPosn**

PLS.#.Status will be active when the selected source position is between the PLS.#.OnPosn and the PLS.#.OffPosn. The terms On and Off assume you are traveling in a positive direction. Assume that the PLS.#.Direction is set to "Both". When traveling in the positive direction and the position feedback reaches the OnPosn, the PLS.#.Status will activate. As the motor continues in the same direction, the PLS.#.Status will deactivate when feedback position reaches or exceeds the OffPosn. If motor travel changes to the negative direction, the PLS.#.Status will activate when position feedback reaches the OffPosn, and will

deactivate when it continues past the OnPosn. The important thing to remember is that the PLS.#.Status will be active if between the PLS On and Off points.

If using negative values for the OnPosn and OffPosn, the most negative value should go in the OnPosn parameter, and the least negative value should go in the OffPosn.

If the PLS has a rollover point, and the OnPosn is greater than the OffPosn, the PLS will be active whenever the axis is not between the On and Off positions, and inactive whenever the axis is between the two positions. However, the PLS.#.Status will not turn on until it reaches the OnPosn the first time.

PLS On Point **PLS.#.OnPosn**

PLS.#.Status will be active when the selected source position is between the PLS.#.OnPosn and the PLS.#.OffPosn. The terms On and Off assume the motor is traveling in a positive direction. Assume that the PLS.#.Direction is set to "Both". When traveling in the positive direction and the position feedback reaches the OnPosn, the PLS.#.Status will activate. As the motor continues in the same direction, the PLS.#.Status will deactivate when feedback position reaches or exceeds the OffPosn. If motor travel changes to the negative direction, the PLS.#.Status will activate when position feedback reaches the OffPosn, and will deactivate when it continues past the OnPosn. The important thing to remember is that the PLS.#.Status will be active if between the PLS On and Off points.

If using negative values for your OnPosn and OffPosn, the most negative value should go in the OnPosn parameter, and the least negative value should go in the OffPosn.

If the PLS has a rollover point, and the OnPosn is greater than the OffPosn, the PLS will be active whenever the axis is not between the On and Off positions, and inactive whenever the axis is between the two positions. However, the PLS.#.Status will not turn on until it reaches the OnPosn the first time.

PLS Enable **PLS.#.PLSEnable**

This destination is used to enable an individual PLS. A PLS can be enabled though the Assignments view in PowerTools FM3 or from a program. If enabled, the PLS will begin to function as soon as the drive has been homed or a DefineHome destination has been activated. Master Posn Valid must be active (Master Define Home is activated) if using a master signal for PLS source.

PLS Rollover Enable **PLS.#.RotaryRolloverEnable**

This parameter is used to enable the RotaryRolloverPosn for the individual PLS.

PLS Rollover Position **PLS.#.RotaryRolloverPosn**

This parameter is the absolute position of the first repeat position for this PLS. When enabled it causes the PLS to repeat every time this distance is passed. The repeating range begins at

an absolute position of zero and ends at the RotaryRolloverPosn. For example in a rotary application a PLS could be setup with an OnPosn of 90 degrees and an OffPosn of 100 degrees. If the RotaryRolloverPosition is set to 360 degrees the PLS would come on at 90, go off at 100, go on at 450 (360+90), go off at 460 (360+100), go on at 810 (2*360+90), go off at 820 (2*360+100), and continue repeating every 360 degrees forever.

PLS Source **PLS.#.Source**

PLSs can be assigned to three different sources: MotorPosnFeedback, MotorPosnCommand, or MasterPosnFeedback. This parameter determines which position signal the PLS uses to reference its OnPosn and OffPosn in order to determine its PLS.#.Status parameter.

PLS Status **PLS.#.Status**

This source is active when the position of the PLS source (motor or master) is greater than or equal to the OnPosn and less than the OffPosn.

Positive Direction **PositiveDirection**

This bit is used to select which direction of motor rotation is considered motion in the positive direction. Select from CW or CCW.

Position Command **PosnCommand**

Position command sent to the EN drive by the FM-3. This parameter does not take following error into account. See also PosnFeedback and FollowingError. Units are in user units.

Position Error Integral Enable **PosnErrorIntegralEnable**

This parameter is used to enable the position error integral compensation. See also Position Error Integral Time Constant.

Position Error Integral Time Constant **PosnErrorIntegralTimeConstant**

Position Error Integral parameter is a control term, which can be used in Pulse mode to compensate for the continuous torque required to hold a vertical load against gravity or to minimize following error.

The user configures this control term using the “Position Error Integral Time Constant” parameter. This parameter determines how quickly the drive will correct for in-position following error. The time constant is in milliseconds and defines how long it will take to decrease the following error to 37 percent of the original value. In certain circumstances the value actually used by the drive will be greater than the value specified here.

Min Time Constant = 1000/Response

For example, with “Response” set to 50, the minimum time constant value is 1000/50 = 20 msec.

Position Feedback **PosnFeedback**

Feedback position is the actual motor position in user units. PosnCommand minus the PosnFeedback is the FollowingError

Position Feedback In Counts **PosnFeedbackInCounts**

Motor encoder position in encoder counts since power up. This position reflects the feedback position of the motor and is not scaled into user units. It can be used to confirm the exact position of the motor in applications where precise positioning is required.

Power Stage Enabled **PowerStageEnabled**

This source (output function) is active when the drive's power stage is enabled.

PowerUpCount

Number of times the drive has been powered up since it was manufactured.

PowerUpTime

Time elapsed since last drive power-up. Units is minutes.

PowerUpTimeTotal

Total power up time since the drive was manufactured. Units is hours.

Any Complete **Program.AnyComplete**

This source is activated when any program ends normally. If a program ends due to a fault or the stop destination, this source does not activate. Deactivates when any program is initiated.

Initiate **Program.#.Initiate**

When activated, this destination initiates the specified program unless a program, index, home, or jog is already executing or a stop is active.

Name **Program.#.Name**

This is a character string which the user can assign to an individual program. It allows the user to give a descriptive name to programs for ease of use.

Program Complete **Program.#.ProgramComplete**

This source is activated when a specific program ends normally. If the program ends due to a fault or the stop destination, this source does not activate. Deactivates when the specific program is initiated again.

Response

The Response parameter adjusts the velocity loop bandwidth with a range of 1 to 500 Hz. In general, it affects how quickly the drive will respond to commands, load disturbances and velocity corrections. A good value to start with (the default) is 50 Hz.

Rotary Rollover Enable **RotaryRolloverEnable**

This parameter is used in applications with a predefined repeat length. One example would be a rotary table with a rotary rollover position of 360 degrees. The position will rollover to zero when the axis position gets to 360 degrees. (358, 359, 359.999, 0.0000, 1, 2, and so on.) The rollover point is defined to be exactly the same position as 0.

Selector Input Destinations **Selector.SelectLinesUsed**

The selector is a binary to decimal decoder. This parameter selects the number of destinations (input lines) to be used by the selector. The number of lines used determines the number of sources (selections) that can be made by the selector; that is 2 input lines can select 4 destinations (selections), 5 input lines can select 32 destinations. Range is 1 to 8.

Select **Selector.#.Select**

This source selects Binary inputs to the selector, usually assigned to input lines. This is level sensitive.

Selection **Selector.#.Selection**

This source selects Decimal outputs from the selector, assigned to indexes, homes or programs.

Initiate **Selector.SelectorInitiate**

When this destination is activated, the selector checks the status of all Selector.Select destinations to determine which Selector.Selection to activate.

ShuntActive

This source is active when the drive's internal shunt is active (conducting current).

ShuntPowerRMS

This parameter models the thermal heating and cooling of the drive internal shunt. This parameter indicates the percent of shunt capacity utilization and is based on the Heat Sink RMS value. When this value reaches 100 percent the drive will generate an RMS Shunt Power Fault. This is only applicable to the EN-208 and EN-214.

Enable Software Travel Limits **SoftwareTravelLimitEnable**

Software travel limits can be used to limit machine travel. They are often setup inside the hardware travel limits to add a level of protection from exceeding the machines travel limits. The SoftwareTravelLimitMinusActive source (output function) is active when the SoftwareTravelLimitMinusPosn is reached or exceeded. Motion is halted using the TravelLimitDecel whenever a hardware or software travel limit is hit or exceeded. Software travel limits are not active unless Absolute Position Valid is active.

Software Travel Limit Minus Activate **SoftwareTravelLimitMinusActive**

The SoftwareTravelLimitMinusActive source is active when the SoftwareTravelLimitMinusPosn is reached or exceeded. Motion will come to a stop using the TravelLimitDecel ramp. Software travel limits are not active unless enabled and Absolute Position Valid is active.

Software Travel Limit Plus Activate **SoftwareTravelLimitMinusPosn**

The SoftwareTravelLimitMinusActive source will activate when the SoftwareTravelLimitMinusPosn is reached or exceeded. Motion will come to a stop using the TravelLimitDecel. Software travel limits are not active unless enabled and Absolute Position Valid is active.

Software Travel Limit Plus Active **SoftwareTravelLimitPlusActive**

The SoftwareTravelLimitPlusActive source is active when the SoftwareTravelLimitPlusPosn is reached or exceeded. Motion will come to a stop using the TravelLimitDecel ramp. Software travel limits are not active unless enabled and Absolute Position Valid is active.

Software Travel Limit Plus Position **SoftwareTravelLimitPlusPosn**

The SoftwareTravelLimitPlusActive source is active when the SoftwareTravelLimitPlusPosn is reached or exceeded. Motion is halted using the TravelLimitDecel whenever a hardware or software travel limit is hit or exceeded. Software travel limits are not active unless enabled and Absolute Position Valid is active.

Stop

Activate this destination to stop all motion and programs. If Stop is activated when a Jog, Index, Home or Program is in progress, they will decelerate to zero speed at the Stop Decel ramp. When Stop is active, all Jog, Home, Index and Program initiate destinations will be ignored. When it is deactivated, all level sensitive and active input functions (Jog.0.PlusActivate, Jog.0.MinusActivate, etc.) will become operational. For example, if the Jog.PlusActivate input function is active when the Stop input function is deactivated, the Jog.Plus motion will initiate using the acceleration found in the Jog.0.Accel parameter. This is level sensitive.

Stop Deceleration **StopDecel**

Deceleration rate used when the Stop destination is activated.

Torque Command **TorqueCommand**

This parameter is the torque command value before it is limited. The torque command may be limited by either the Torque Limit (if the Torque Limit Enable destination is active) or current foldback.

Limited Torque Command **TorqueCommandLimited**

This is the actual torque commanded to the motor. This value is the result after the TorqueCommand is limited by the current foldback or the TorqueLimit value (if enabled).

Torque Level **TorqueLevel**

This parameter is compared to the TorqueCommand. If the absolute value of the TorqueCommand is greater than or equal to the TorqueLevel the TorqueLevelActive source is activated. This parameter is specified in Torque User Units.

Torque Level Active **TorqueLevelActive**

This source is used to indicate that the absolute value of the TorqueCommand is greater than or equal to the TorqueLevel setting.

Torque Limit	TorqueLimit

This is the level to which the TorqueCommand will be limited when the TorqueLimitEnable input function is active.

Torque Limit Active	TorqueLimitActive

Active when the TorqueCommand is greater than the TorqueLimit and the TorqueLimitEnable input function is active.

Torque Limit	TorqueLimitEnable

This destination is used to enable the TorqueLimit. This is level sensitive.

Travel Limit Minus Activate	TravelLimitMinusActivate

This destination is used to activate the travel limit minus fault. It should be assigned to the travel limit minus sensor. When it is activated the drive will decelerate to a stop using the deceleration rate defined in the TravelLimitDecel parameter. This is level sensitive.

Travel Limit Plus Activate	TravelLimitPlusActivate

This destination is used to activate the travel limit plus fault. It should be assigned to the travel limit plus sensor. When it is activated the drive will decelerate to a stop using the deceleration rate defined in the TravelLimitDecel parameter. This is level sensitive.

Travel Limit Deceleration	TravelLimitDecel

This parameter defines the ramp used to decelerate the motor to a stop when any travel limit is activated

Travel Limit Minus Active	TravelLimitMinusActive

This source is active when the TravelLimitMinusActivate is active.

Travel Limit Plus Active	TravelLimitPlusActive

This source is active when the TravelLimitPlusActivate is active.

Decimal Places	TorqueUnits.Decimal

This parameter is the decimal point location for user torque units.

Units Name **TorqueUnits.Name**

The User can specify a torque unit name of up to 12 characters. Default is % Cont..

Percent Continuous **TorqueUnits.PercentContinuousCurrent**

This parameter is the denominator of torque scaling factor. This is an amount of continuous current in percent that is equal to the TorqueUnits.Torque parameter.

Torque **TorqueUnits.Torque**

This parameter is the numerator of the torque scaling factor. This is an amount of torque in Torque User Units that is equivalent to one unit of PercentContinuousCurrent (denominator of scaling factor). This scaling factor is used to relate actual current or torque to user units.

Velocity Command **VelCommand**

The Velocity Command is the velocity that the FM-3 drive is commanding the motor to run at. This command is generated by the drive velocity control loop. It is displayed in user units.

Velocity Feedback **VelFeedback**

This is the feedback (or actual) velocity. It will always return the actual motor velocity, even in synchronized applications in which the master axis is halted during a move.

Decimal Places **VelocityUnits.Decimal**

This parameter is used to select the number of decimal places used in velocity units scaling. Using a high number of decimal places will improve velocity resolution, but will also limit the maximum velocity. This parameter is selectable between 0 and 6 decimal places. The number of decimal places set in this parameter determines the number of decimal places used in all velocity parameters throughout the PowerTools FM-3 software.

Scaling **VelocityUnits.ScalingFlag**

This parameter enables separate velocity and distance user units, name and scaling.

VelocityUnits.DistVelScale

Velocity units can be scaled to different from distance units, i.e. user distance units are inches and velocity units are feet per minute, instead of inches per minute. To do this, simply enter 12 to set 1 foot equal to 12 inches (1 velocity unit = 12 distance units). Range is 1 to 1000, integers only.

Units Name	VelocityUnits.Name

If the user wants the velocity units to have a different distance scaling than the distance units a name can be entered here up to 12 characters, e.g. user distance units are inches and velocity units are feet per minute.

Time Scale	VelocityUnits.TimeScale

Velocity time scale can be set to user units per second or user units per minute, used for all real-time velocities throughout the PowerTools FM-3 software.

Parameters Saved to NVM

The following parameters will be saved to Non-Volatile Memory (NVM). See the How Communications Work section of the Operational Overview chapter for more details.

- AccelerationType
- DefineHomePosition
- FeedholdDecelerationTime
- FeedrateOverride
- FollowingErrorLimit
- StopDeceleration
- TorqueLevel
- Torque Limit
- TravelLimitDeceleration
- Home.#.Acceleration
- Home.#.Deceleration
- Home.#.LimitDistance
- Home.#.Position
- Home.#.SpecifiedOffset
- Home.#.Velocity
- Index.#.Acceleration
- Index.#.Deceleration
- Index.#.Distance
- Index.#.PLSEnable

- Index.#.PLSOffDistance
- Index.#.PLSOnDistance
- Index.#.RegistrationOffset
- Index.#.RegistrationWindowEnd
- Index.#.RegistrationWindowStart
- Index.#.Velocity
- Jog.#.Acceleration
- Jog.#.Deceleration
- Jog.#.Velocity
- MasterDistUnits.CharacteristicDistance
- MasterDistUnits.Counts
- MasterEncoder.DefineHomePosition
- PLS.#.OffPosition
- PLS.#.OnPosition
- PLS.#.RotaryRolloverPosition

Installation

Basic Installation Notes

You are required to follow all safety precautions during start-up such as providing proper equipment grounding, correctly fused power and an effective Emergency Stop circuit which can immediately remove power in the case of a malfunction. See the “Safety Considerations” section for more information.

Mechanical Installation

The FM-3 detects and verifies the drive serial number when its attached to an E Series drive. If an FM-3 is moved from one drive to another, it will detect the difference in serial numbers and generate an Invalid Configuration fault.

Two aligning tabs, a locking latch and a 100-pin connector are used to attach the FM-3 to an E Series drive. All electrical connections between the FM-3 and the E Series drive are accomplished with the single connector located on the rear of the FM-3.

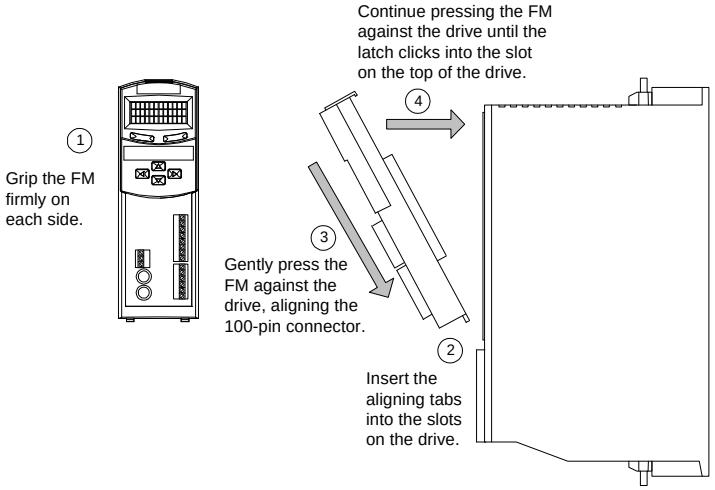

Continue pressing the FM against the drive until the latch clicks into the slot on the top of the drive.
4
1
Grip the FM firmly on each side.
3
Gently press the FM against the drive, aligning the 100-pin connector.
2
Insert the aligning tabs into the slots on the drive.

Figure 54: Attaching the FM-3 to an E Series Drive

Caution

Do not attach or detach the FM-3 when power is applied to the drive. It could cause drive instability.

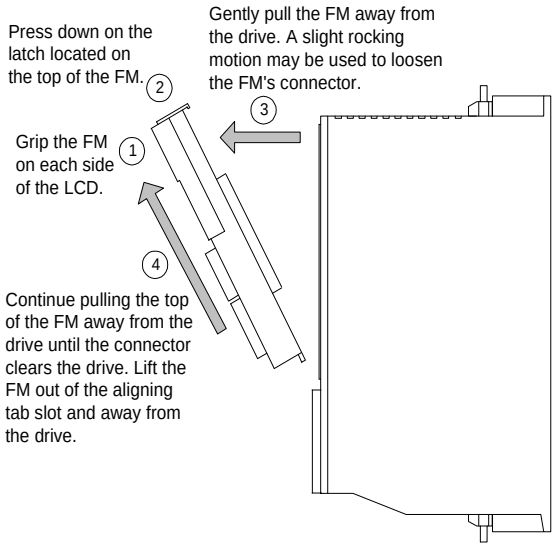

Press down on the latch located on the top of the FM.
2
Gently pull the FM away from the drive. A slight rocking motion may be used to loosen the FM's connector.
3
Grip the FM on each side of the LCD.
1
4
Continue pulling the top of the FM away from the drive until the connector clears the drive. Lift the FM out of the aligning tab slot and away from the drive.

Figure 55: Detaching the FM-3 from the Drive Connections

Caution

Do not attach or detach the FM-3 when power is applied to the drive It could cause drive instability.

Modbus Communications

The drive's serial communication protocol is Modbus RTU slave with a 32 bit data extension. The Modbus protocol is available on most operator interface panels and PLC's.

Serial Communications Specifications	
Max baud rate	19.2k (default)
Start bit	1
Stop bit	2
Parity	none
Data	8

EMERSON Motion Control's Motion Interface panels are supplied with a Modbus master communications driver.

Software Installation

Emerson Motion Control PowerTools® FM-3 software was designed for users who need access to all setup options and diagnostic information for the drive. The software provides access to all commonly used drive parameters.

PC System Configuration

Required:

- Microsoft® Windows® 95/98 or Microsoft Windows NT® 4.0 (service pack 3)
- Serial port
- Mouse
- 3.5" floppy drive for installation

Recommended:

- 16 MB or more RAM for Windows 95/98, and 32 MB or more for Windows NT.
- Up to 45 MB of hard disk space required for a full installation; 15 MB for minimum installation.

Optional:

- Windows-compatible printer
- Modem and Internet access

Preparing for PowerTools FM-3 Installation

- Close all programs and turn off virus protection software to prevent installation conflicts.
- If you have another version of PowerTools on your computer and do not want to overwrite that version, change the default installation folder for PowerTools during installation setup.
- To avoid overwriting programs you have customized or created, place them into a separate folder before you install.

Installing PowerTools FM-3

You must be running Microsoft Windows 95/98 or Microsoft Windows NT 4.0 to install M-3.

To install PowerTools:

1. Insert the PowerTools disk 1 into the floppy drive.
2. From the Start menu, choose **Run**.

3. In the Run dialog box, type *a:/setup*, where *a* is the letter assigned to your 3.5" floppy drive.

4. Click the *OK* button, then follow the instructions on your screen.

Installation will take about five minutes. The install program automatically:

- Creates the directory on your hard drive named:

  C:\EMERSON\PT-FM3

- Creates a new Windows group called "EMERSON Motion Control-FM3".
- Loads the required DLL's into your Windows system directory.
- Loads PowerTools into the EMERSON\PT-FM3 directory.
- Loads drive setup files into the EMERSON\APPS directory.

Note

To complete the installation for Windows 98 and NT, you may be required to restart Windows.

Refer to the readme.txt file located on the installation floppy for more information.

If You Need Help with Installation

If you encounter problems while installing PowerTools, call Technical Support at (952)995-8033 (24 hour assistance).

Starting and Exiting PowerTools

Starting the Software

1. Click the Start button on the taskbar at the bottom left of the screen.

2. Click on the Programs selection. The Programs menu appears.

3. Click on the "EMERSON Motion Control" folder icon.

4. Click on the "PowerTools" folder icon.

5. Click on the PowerTools program icon.

Exiting the Software

Choose **File** from the menubar, then click Exit. If you are using the keyboard, press the shortcut key combination Alt+F, X.

If you made changes, PowerTools displays an Alert dialog box asking whether you want to save your current work. Click the *Yes* button or press Enter to save your work, or click the *No* button to quit without saving.

Accessing Help

All PowerTools tabs and dialog boxes provide a *Help* button. This retrieves help for that particular subject and provides details to assist you in the use of each tab or dialog box. Also, provided is a index-based help system. There is the ability to search the online help using a keyword. Lastly, PowerTools has context sensitive help capability.

Online Help displays detailed procedural and reference information. It contains some information that does not appear in this manual.

To display Help in PowerTools, do one of the following:

- Press F1.
- Choose a command from the Help menu.
- Click the *Help* button on a dialog box or tab.
- Click the Help icon on the toolbar.
- Click the Context Sensitive Help button on the toolbar and then the field on the tab.

Quick Start

The quick start guide provides information on the basic operation and functions of the FM-3. The quick start steps cover only the most basic steps required to setup a FM-3 for use with a drive.

Basic Setup Steps

Step 1: Opening a New Configuration Window

To open a new configuration window, click either on the New File icon at the left end of the Emerson Motion Control PowerTools® FM-3 toolbar, or click on File-New from the pulldown menu as shown in Figure 56:.

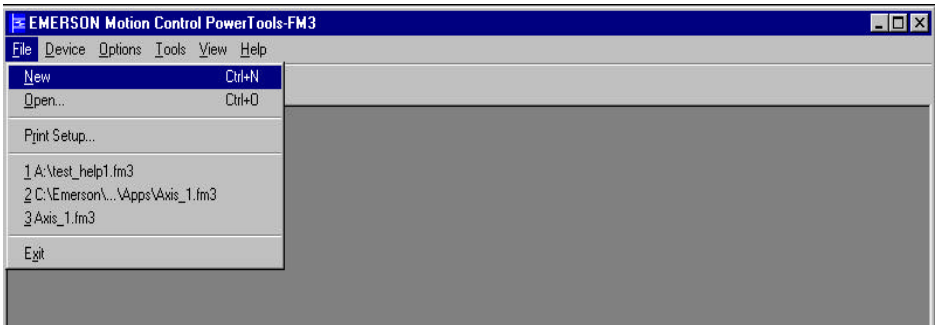


Figure 56: Opening a new Configuration

This opens a new offline configuration window allowing you to setup all FM-3 parameters. The communication status remains “offline” until the configuring is complete and you download the application to the FM-3.

Step 2: Enter All Setup Data

General Setup View

The FM-3 software view is broken up into two areas. The left side view contains the configuration hierarchy window, which acts as a navigation frame. Some elements in the hierarchy can be expanded to show more detailed setup views. Elements with a “+” icon to the left of them may be expanded for added detail.

To expand the element, place your mouse pointer over the “+” symbol and click on it using the left mouse button. The group then expands to show the setup views underneath it. To collapse the expanded group, click on the “-“ symbol located to the left of the group heading.

To select one of the individual setup views, click on the element in the hierarchy view. The element will then be highlighted, and the right side of the view will show all setup parameters for that subject. Figure 2 shows the Setup view selected.

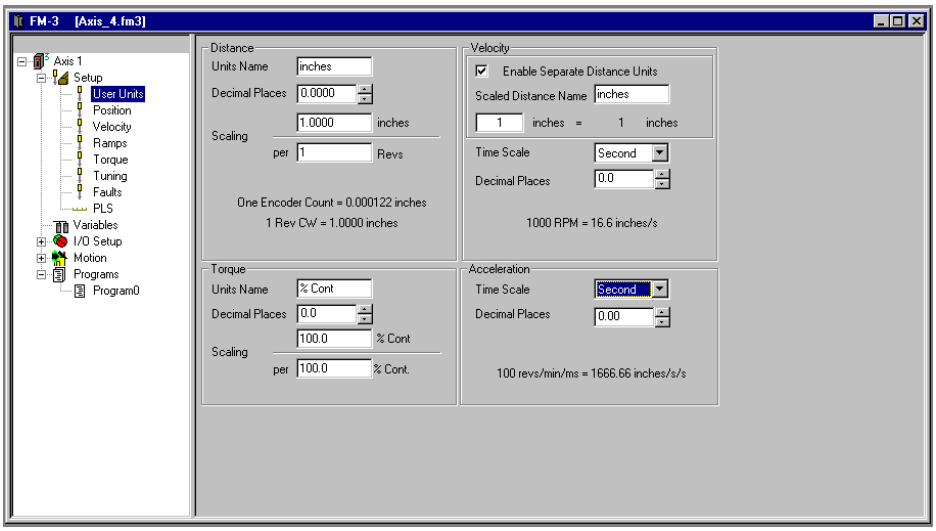


Figure 57: General Setup View

On this view, you must enter the following parameters before entering other setup views:

Drive Type – Select, from the list box, the type of drive to which the FM-3 is attached.

Motor Type – Select, from the list box, the type of motor being used in this application.

Line Voltage – Select, from the list box, the line voltage applied (115VAC or 230VAC).

Target Drive Address – Enter the Modbus address for this drive/FM-3 system. PowerTools FM-3 will download the configuration to that specific drive only.

Please take the time to check that these parameters are correct.

User Units View

Determine which types of units the drive should use to measure motion and enter them on this view. Figure 3 shows the Setup group in the hierarchy expanded with the User Units element selected.

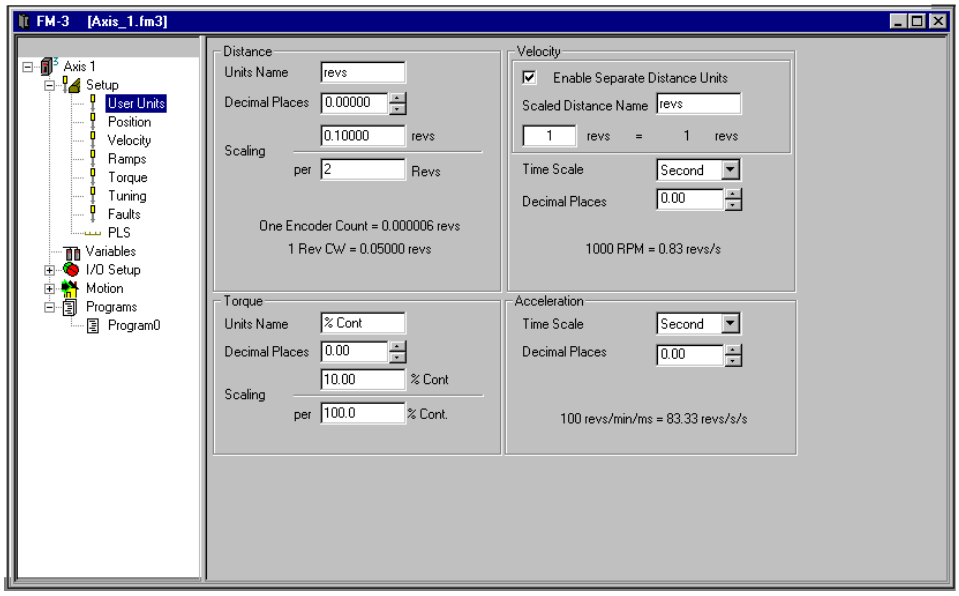


Figure 58: User Units View

Now setup the parameters in the Distance group on the User Units view. The Tab key is used to easily jump through the available parameters. After entering data in a field, press the Tab key to move to the next parameter. The parameters you must fill out are as follows:

Units Name – Enter the character string you wish to use for your distance units.

Decimal Places – Enter the number of digits after the decimal place you wish to use in all distance parameters throughout the software (i.e. index distances/positions, PLS On/Off points). Use the spinner window Up/down buttons to increase or decrease the number of digits.

Distance Units Scaling – Enter the number of User Units the motor/load travels for each revolution of the motor. You must enter the numerator and the denominator for this scaling factor.

Master Units View

Use this view if you wish to set parameters for synchronized motion.

Position View

The Settings group on this view sets the Define Home Position.

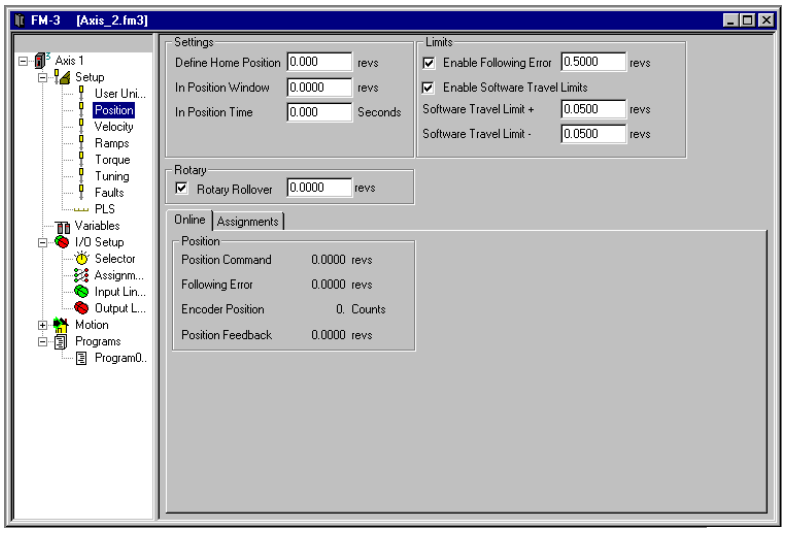


Figure 59: Position View

To setup following error limit and software travel limits, fill out the following parameters in the Limits Group:

Enable Following Error – This checkbox enables (when checked) and disables the following error fault capability. If checked, a following error fault is generated whenever following error exceeds the value entered in the parameter box. If unchecked, a following error fault will never be generated.

Following Error Limit – If Following Error meets or exceeds this value, a Following Error Fault will be generated.

Enable Software Travel Limits - This checkbox enables (when checked) and disables the software travel limits. If enabled, the software travel limits are not active unless the Absolute Position Valid Source is active. (For more information please see Software Travel Limits information in the Setting Up Parameters section of the Reference Manual.)

Software Travel Limit + and - - If the position feedback ever exceeds these values when traveling in a positive or negative direction, the motor will come to a stop at the Travel Limit Decel rate defined on the Ramps view.

Velocity View

This view sets up the feedrate override velocity. If your application does not call for an overall scaling factor, leave this at the default of 100%.

Ramps View

You must setup the deceleration rate to be used on the Ramps view if a travel limit is encountered.

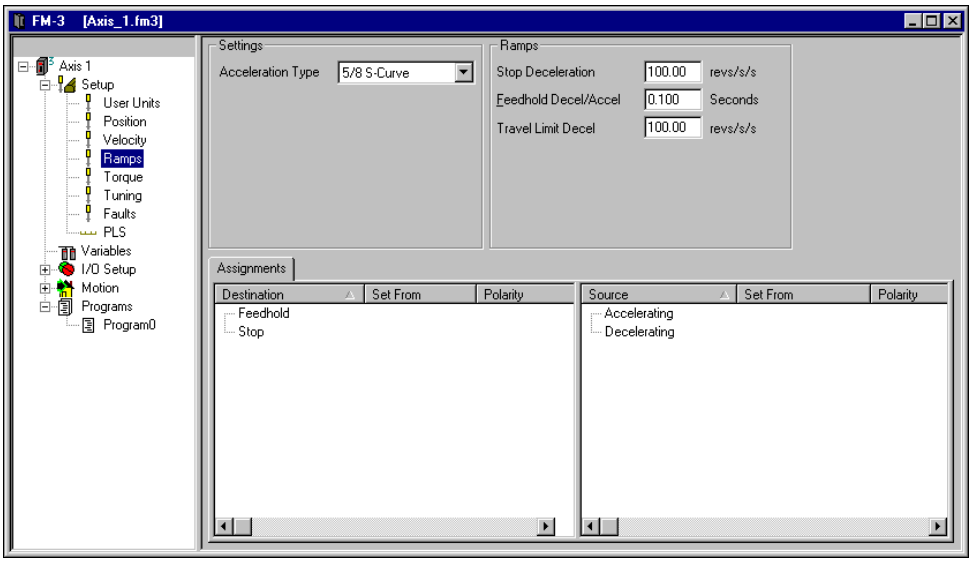


Figure 60: Ramps View

Travel Limit Decel – The deceleration ramp is used if either a Software or Hardware Travel Limit is encountered. The deceleration units of the ramp are defined in the User Units setup view under the Acceleration group.

Torque View

If your application calls for specific torque levels and limits, use the parameters on this view.

Tuning View

EN Series Drives are designed to handle a 10:1 inertia mismatch without a tuning requirement. If you have calculated and know the inertia mismatch, the value can be entered on the Tuning view for ideal drive response and velocity regulation.

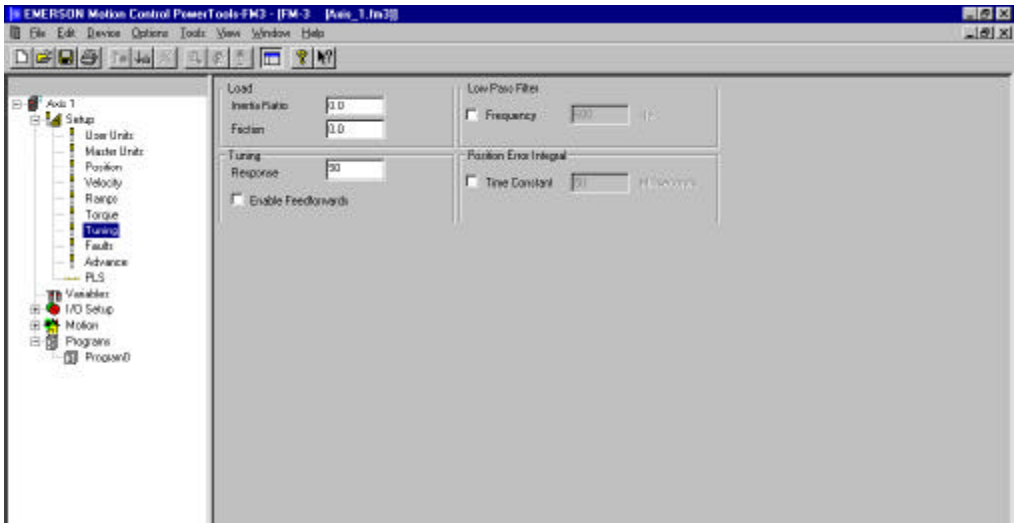


Figure 61: Tuning View

Inertia Ratio - This parameter is the ratio between the reflected inertia of the load and the inertia of the motor rotor. For assistance in calculating the Inertia ratio, see the Determining Tuning Parameter Values section of the FM-3 Reference Manual.

PLS View

Use this view to setup Programmable Limit Switches (PLS), if your application requires them. They are assigned in the same way as the Assignments views in Step 3 of this guide.

Step 3: Making Assignments

The Assignments view is found by expanding the I/O Setup group in the hierarchy. The Assignments define how the system operates. Sources, located on the left side of the view, are functions or events that activate based on drive/motor activity. Destinations, located on the right side of the view, are functions that need to be triggered or activated (i.e. Index Initiate, Program Initiate, etc.).

The example application requires initiating three indexes using three separate hardware inputs. Therefore, the Input sources need to be assigned to the Index Initiate destinations.

To make an assignment, position your mouse pointer over the source you wish to assign to a destination, click and hold the left mouse button. While still holding the button, drag your pointer over the destination where you wish to assign this source, and release the button. The “Assigned To” and the “Set From” columns should then reflect the assignment that has been made. Figure 7 shows how these assignments are made.

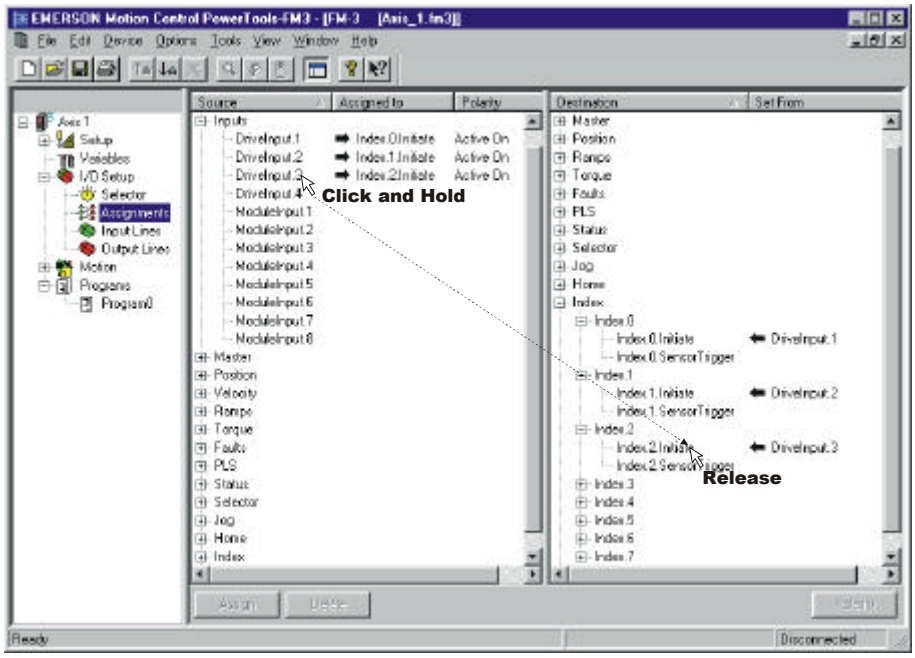


Figure 62: Making Assignments

Notice that a Name can be associated to each input line. Each input line can also have a debounce time.

Name – This 12-character string allows a descriptive name assigned to an input line. This makes the configuration easier to follow.

Debounce – This is the minimum time a specific input must be on, for recognition by the drive/FM-3. This helps prevent false triggering for applications in electrically noisy environments. The units for this parameter are seconds, with resolution of 0.001 seconds.

The Output Lines view shows assignments that have been made to the Outputs group of Destinations in the assignment view. Assignments to this view are made in the same way as assignments to the Input Lines view. After all assignments have been made, the Output Lines view may look like Figure 8.

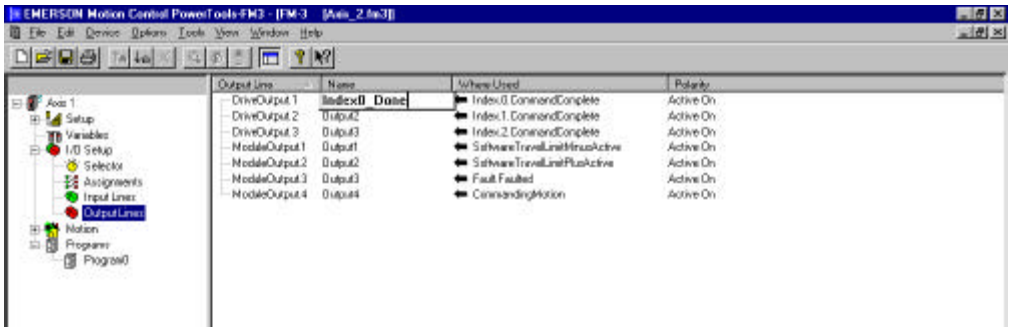


Figure 63: Output Lines View

A name can also be given to each hardware output line. Figure 8 shows how to enter a name for an individual output line.

Name – This 12-character string allows a descriptive name assignment to an output line. This makes the configuration easier to follow.

Step 4: Setting Up Motion Parameters

Now that all of the assignments have been made, you next setup the jog, home, and index definitions. To do this, it is necessary to expand the Motion group in the hierarchy window.

Jog Setup

The Jog views allow you to setup any jogs required by your application. Select either the Jog0 or Jog1 to enter the specific Jog parameters. Figure 11 shows Jog1 setup view (renamed Jog Fast). Once in the individual jog views, the up and down arrows next to the Jog Number parameter can be used to scroll between the jog views (see Figure 9). The following jog parameters have been defined on this view:

Jog Name - Jog0 has been named Jog Slow and Jog1 named Jog Fast. This is a 12-character string that allows you to give a descriptive name to a jog profile.

Jog Velocity – Is the target velocity for the Jog profile. If the Jog.Select0 destination is inactive, the Jog0 velocity is the target velocity, if Jog.Select0 is active, then Jog1 velocity is used.

Jog Acceleration – Is the average acceleration rate used when accelerating to the target velocity.

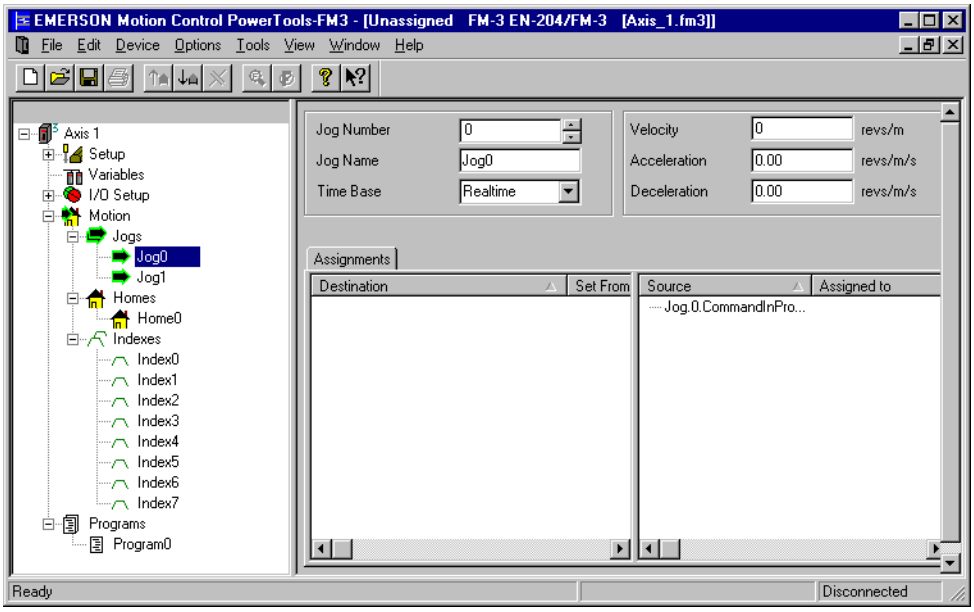


Figure 64: Jog Setup View

Jog Deceleration - Is the average deceleration rate used when decelerating to zero speed or to the new target velocity.

Home Setup

Next, select the Home view from the hierarchy window. Figure 10 shows the Home view and its associated home parameters.

Home Reference – Determines what signal is used as the home reference. The available options are an external sensor, the encoder marker pulse, or both. If sensor is selected, the home profile will travel at its target velocity until the Home.0.SensorTrigger is activated.

Velocity – Is the target velocity for the home move. The home move travels at this velocity, until the home reference is seen and then stops. If the motor overshoots the sensor, a reverse move is initiated. Other functions may cause additional motion.

Acceleration – Is the average acceleration rate used to accelerate to the target velocity.

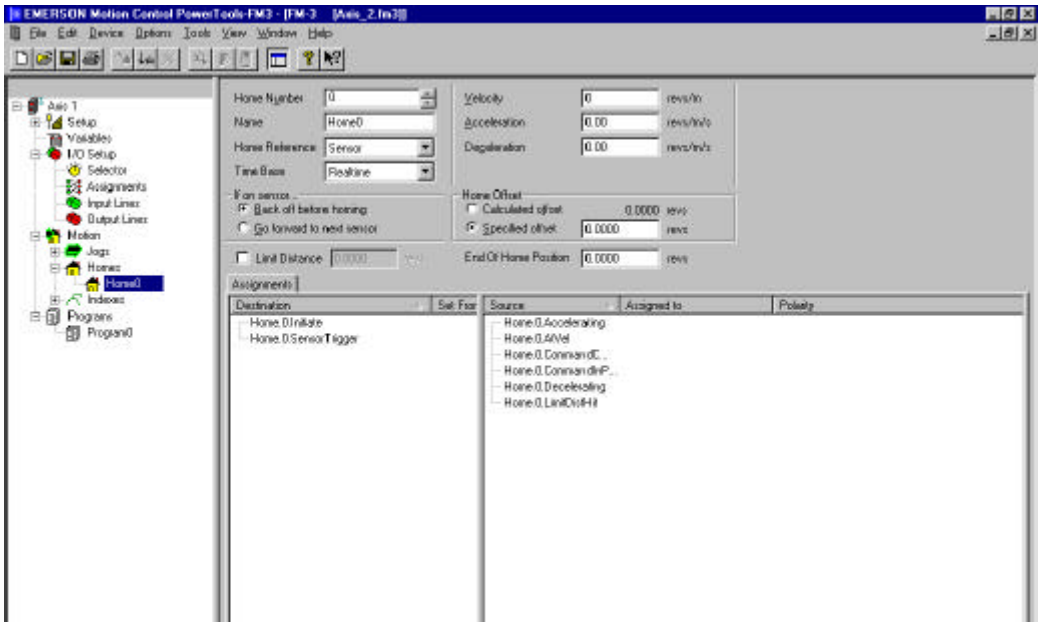


Figure 65: Home Setup View

Deceleration - Is the average deceleration rate used to decelerate from the target velocity to zero velocity at the completion of the home.

Home Offset – By selecting a specified offset, the home will travel at a set velocity until the home reference signal is detected, then travel an additional set distance, before coming to a stop. The deceleration ramp begins at the proper time, to ensure the home completes precisely at the specified offset distance after the home reference was detected. In order to decelerate to a stop immediately upon detecting the home reference, select the "Calculated Offset" radio box.

Limit Distance – Is the maximum incremental distance the home move will travel without seeing a home reference signal. If the home move does travel this distance without seeing a home reference, it will come to a stop exactly at this point, and the Home.0.LimitDistHit source will activate. If the box is left unchecked, the home move will continue without limit, until a home reference is seen.

Index Setup

Next, setup any indexes required by your application. Select one of the indexes from the hierarchy window. Figure 11 shows Index0 (renamed Position A).

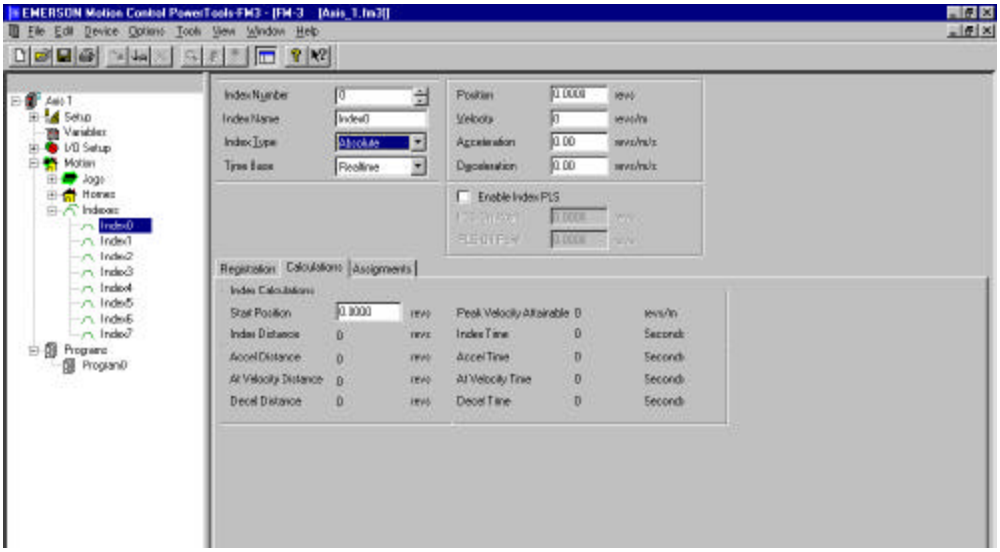


Figure 66: Index Setup View (Absolute Type)

Indexes use the following parameters:

Index Name - Is a 12-character string that gives a descriptive name to an index.

Index Type – Selects the desired type of index from the list box. An Absolute index means the load will travel to an exact position with respect to the home reference point. This index will cause motion in either a positive or negative direction depending on current position of the load with respect to the home reference point. An Incremental index means that the load travels an exact distance from the start of the index.

Position/Distance – If using an absolute index, this is the desired final position after the index is complete. If using an incremental index, then this is the distance you want the load to move from the start of the index. This is a signed value.

Velocity – This is the target velocity of the index profile. The velocity parameter is unsigned and is always greater than zero. Direction of the index is not determined by the velocity, but by the Distance/Position parameter.

Acceleration – Is the average acceleration rate used when accelerating to the target index velocity.

Deceleration - Is the average deceleration rate used when decelerating to zero speed, or to the new target velocity.

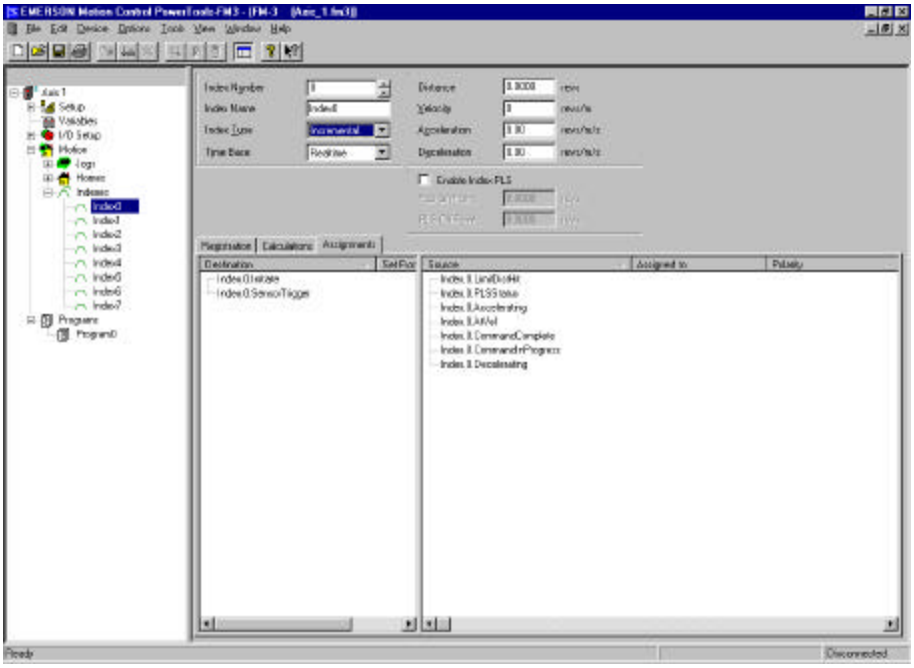


Figure 67: Index Setup View (Incremental Type)

Now that all of the motion parameters are setup, the jogs and indexes can be initiated through assignments to hardware inputs that were made on the assignment view.

Step 5: Creating a Program

Now that the drive setup is complete, assignments are made, and motion profiles are defined, you are ready to create the program. To do this, expand the Program group in the hierarchy window, and select Program 0. Figure 13 shows the programming window and the available instructions.

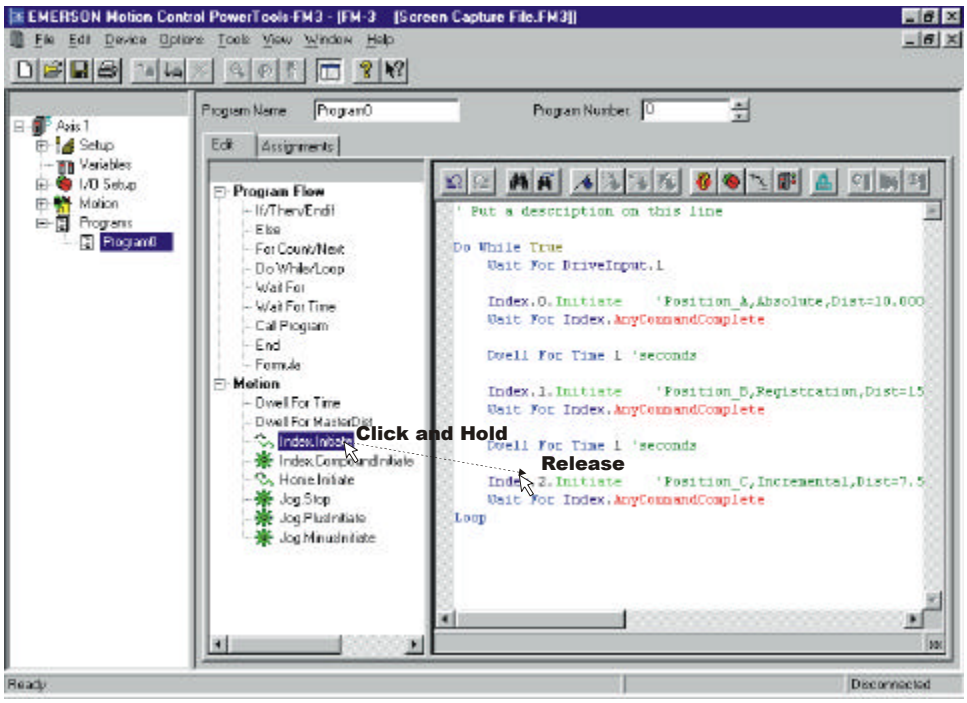


Figure 68: Program View

The window in the middle of the view allows selection from a list of Program Flow and Motion instructions. To insert an instruction into the programming window, you can use the "drag-and-drop" method also used in making assignments. Simply position your mouse pointer over the instruction you wish to use, click the left mouse button and hold it down. While holding the button down, move your pointer into the programming window, and release the mouse button.

To expand the useable area of the programming view, use the Hide/Show Hierarchy on the PowerTools FM-3 toolbar. Figure 14 shows this utility. By clicking on the icon once, the hierarchy window will be hidden, allowing for a larger programming window. To show the hierarchy again, simply click on the icon a second time. Clicking on the icon alternates between showing and hiding the hierarchy view.

The Hide/Show Hierarchy icon can be used on any view in the PowerTools FM-3 software, but will primarily be of use in the Programs view and the Assignments view.

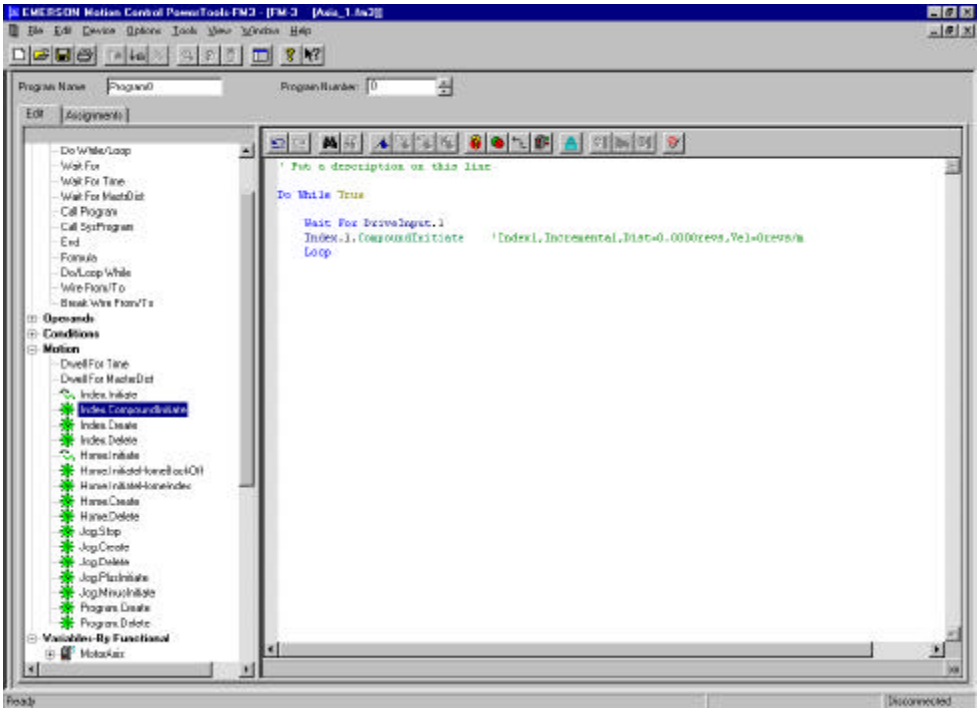


Figure 69: Program View (Expanded)

A comment is automatically inserted after the index instruction, which shows key data about the specific index. Notice comments start with the ‘ apostrophe character.

A “Wait For Index.AnyCommandComplete” instruction is automatically inserted after each Index.Initiate. This ensures that the index has completed before the program continues on to the next line of code.

Once all of the program instructions have been inserted, your program window may look like Figure 14.

Now the program is complete, and is ready to be run. In order to run the program, the drive must be enabled and the setup parameters must be downloaded. Once the download is complete, verify the drive is enabled and initiate your program.

Example Application Start Up

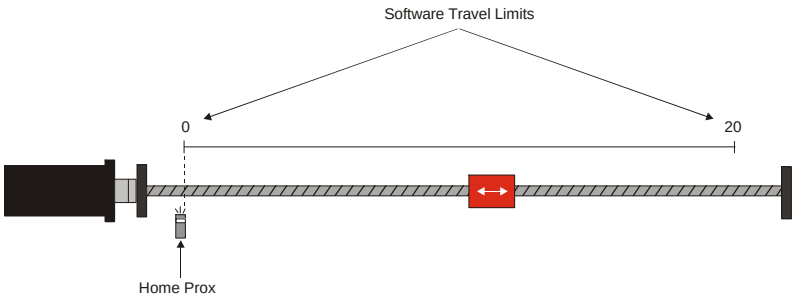


Figure 70: Example System

In order to make the setup easier to understand in this guide, the guide will use an example setup. The example application used in the quick start guide is a ball screw application (see the drawing above).

The example uses these functions:

3 Indexes (Initiated through Assignments or through a Program)

2 Jogs

1 Home (to Sensor)

1 Program

Software Travel Limits

Following Error Limit

Several Setup Views do not require any setup for this example application, and therefore have been skipped. After opening a new Configuration Window and inputting information on the Setup view, move to the User Units view.

User Units View

This example requires use of distance units “Inches” versus the default units “revs”. The example has a linear device that moves 1.00 Inch per 1 motor Revolution. Figure 3 shows the Setup group in the hierarchy expanded with the User Units element selected.

The parameters you must fill out are as follows:

Units Name – Enter the character string you wish to use for your distance units, such as “Inches” for this example.

Decimal Places – Enter the number of digits after the decimal place you wish to use in all distance parameters throughout the software (i.e. index distances/positions). The example

selects three digits allowing resolution of 0.001 Inches on all distance variables. Use the spinner window Up/down buttons.

Distance Units Scaling – Enter the number of User Units the motor/load travels for each revolution of the motor. You must enter the numerator and the denominator for this scaling factor, but for this example, the denominator (bottom portion of the ratio) is at 1.

Position View

The example wishes to limit the Following Error to 0.25 Inches or one motor revolution, and the Software Travel Limits of +20 and 0 Inches.

To setup following error limit and software travel limits, fill out the following parameters in the Limits Group:

Enable Following Error – Enable this checkbox.

Following Error Limit – The example uses a limit of 0.25 Inches.

Enable Software Travel Limits – Enable this checkbox.

Software Travel Limit + and - - If the position feedback ever exceeds these values when traveling in a positive or negative direction, the motor will come to a stop at the Travel Limit Decel rate defined on the Ramps view

Tuning View

This example application has an inertia mismatch of 3.5:1 and has been entered as seen in Figure 6.

Making Assignments

Using this process, three individual assignments are made. The example application wants to turn on an output when each index completes. Therefore, Index Complete sources will be assigned to the hardware Output destinations. Figure 15 shows this process.

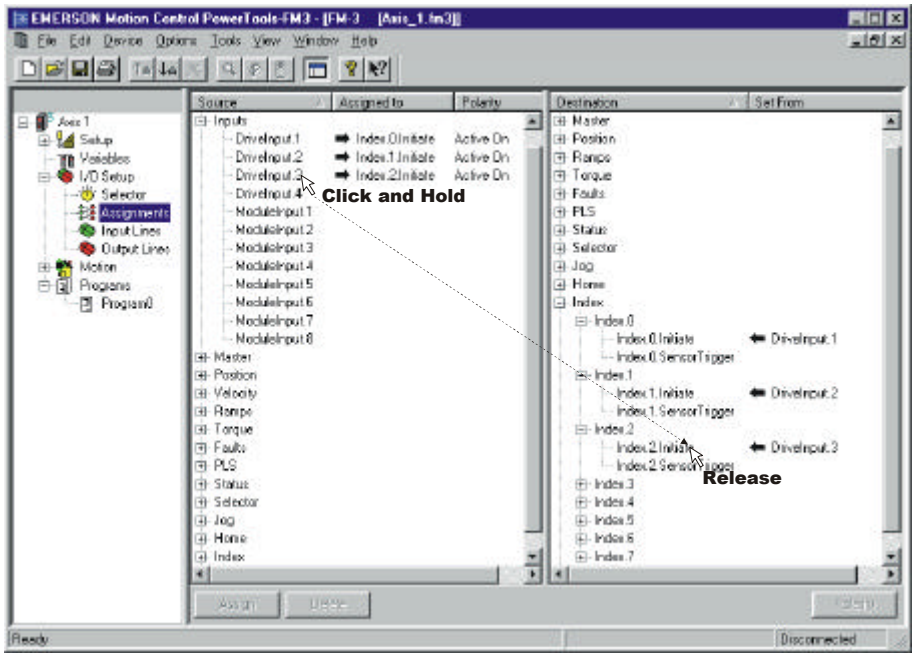


Figure 71: Assignments View

The Input Lines view shows all of the assignments that have been made to the Inputs group of Sources. After all assignments for the example application have been made, the Input Lines view should look like Figure 17.

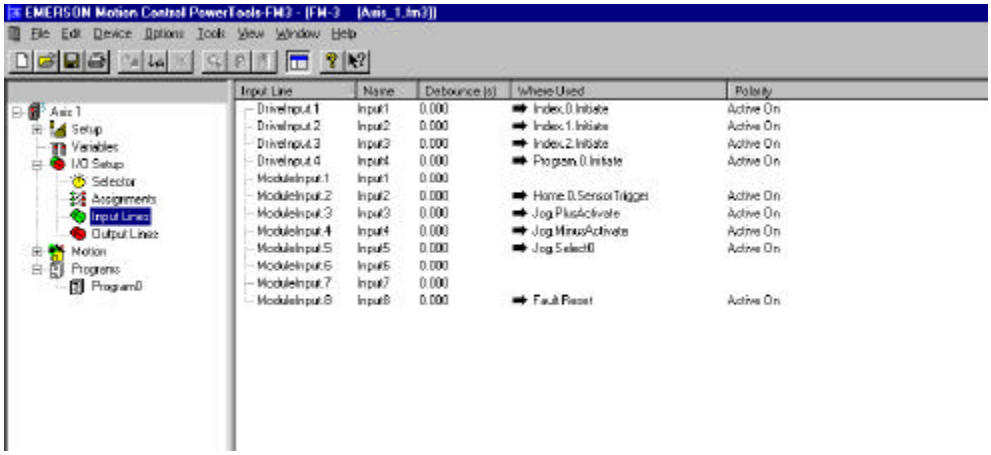


Figure 72: Input Lines View

Notice that a Name can be associated to each input line. Each input line can also have a debounce time. Figure 17 demonstrates how to enter a debounce time for an input line. The following parameters have been filled in for each input line:

The Output Lines view shows assignments that have been made to the Outputs group of Destinations in the assignment view. Assignments to this view are made in the same way as assignments to the Input Lines view. After all assignments have been made for the example application, the Output Lines view will look like Figure 18.

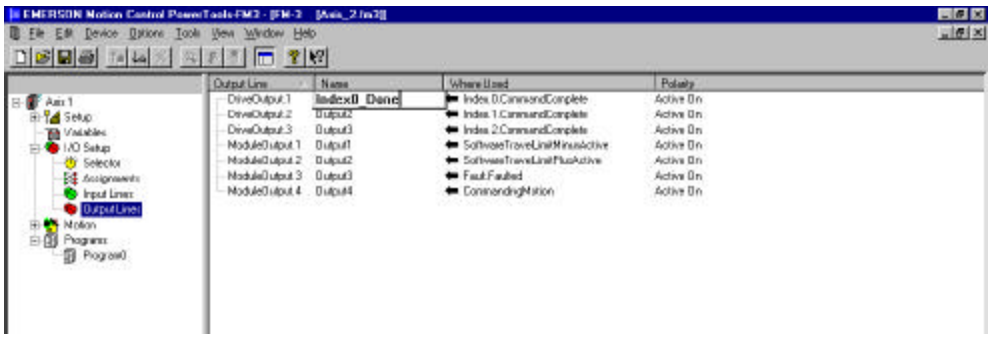


Figure 73: Output Lines View

Home Setup

The example application calls for setup of the following parameters:

Home Reference –The example calls for a Home to Sensor; therefore, “Sensor” is selected as the home reference. Because of this selection, the home profile will travel at its target velocity until the Home.0.SensorTrigger is activated. Figure 16, shows the Home.0.Sensor Trigger has been assigned to ModuleInput.2.

Velocity – The home move travels at this velocity, until the home reference is seen and then stops. If the motor overshoots the sensor, a reverse move is initiated. Other functions may cause additional motion. The example application uses a Home Velocity of 20 inches/min.

Acceleration – Is the average acceleration rate used to accelerate to the target velocity. The example uses an acceleration rate of 100 inches/min/sec.

Index Setup

The example application calls for two Absolute Indexes, and one Incremental Index. Select one of the indexes from the hierarchy window. Figure 19 shows Index0 (renamed Position A).

Absolute Index

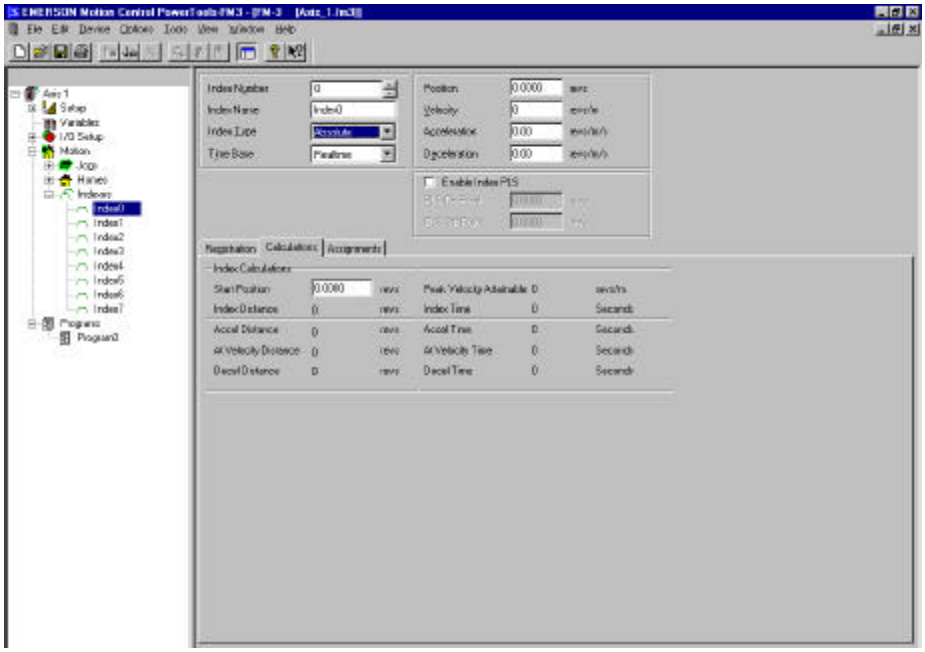


Figure 74: Index Setup View (Absolute Type)

Indexes 0 and 1 are both Absolute type indexes and use the following parameters:

Index Name - Is a 12-character string that gives a descriptive name to an index. Index0 has been named Position A, and Index1 Position B as seen in Figure 18.

Index Type – Select the desired type of index from the list box. Index0 has been defined as an Absolute type of index. This means the load will travel to an exact position with respect to the home reference point. This index will cause motion in either a positive or negative direction depending on current position of the load with respect to the home reference point.

Position – Is a signed value that specifies the absolute position for the load at the completion of the index. For absolute indexes, direction of travel is determined by the starting position of the index. Index 0 has a Position of 10 Inches, and Index 1 has a Position of 0 Inches.

Velocity – Is the target velocity of the index profile. The velocity parameter is unsigned and is always greater than zero. Direction of the index is not determined by the velocity, but by the Distance/Position parameter. Velocity of the indexes in the example are 100 inches/min.

Acceleration – Is the average acceleration rate used when accelerating to the target index velocity.

Deceleration - Is the average deceleration rate used when decelerating to zero speed, or to the new target velocity.

Incremental Index

The following parameters were setup for Index2:

Index Name - Is a 12-character string that gives a descriptive name to an index. Index2 has been named Position C as seen in Figure 19.

Index Type – Select the desired type of index from the list box. Index2 has been defined as an Incremental type of index. This means that the load travels an exact distance from the start of the index.

Distance – The distance is a signed value that specifies the distance the load will travel from the start position. The sign of this parameter determines direction of travel for the load. The distance of Position C has been set to 2.5 inches.

Velocity – Is the target velocity of the index profile. The velocity parameter is unsigned and is always greater than zero. Direction of the index is determined not by the velocity, but by the Distance/Position parameter.

Acceleration – Is the average acceleration rate used when accelerating to the target index velocity.

Deceleration - Is the average deceleration rate used when decelerating to zero speed, or to the new target velocity.

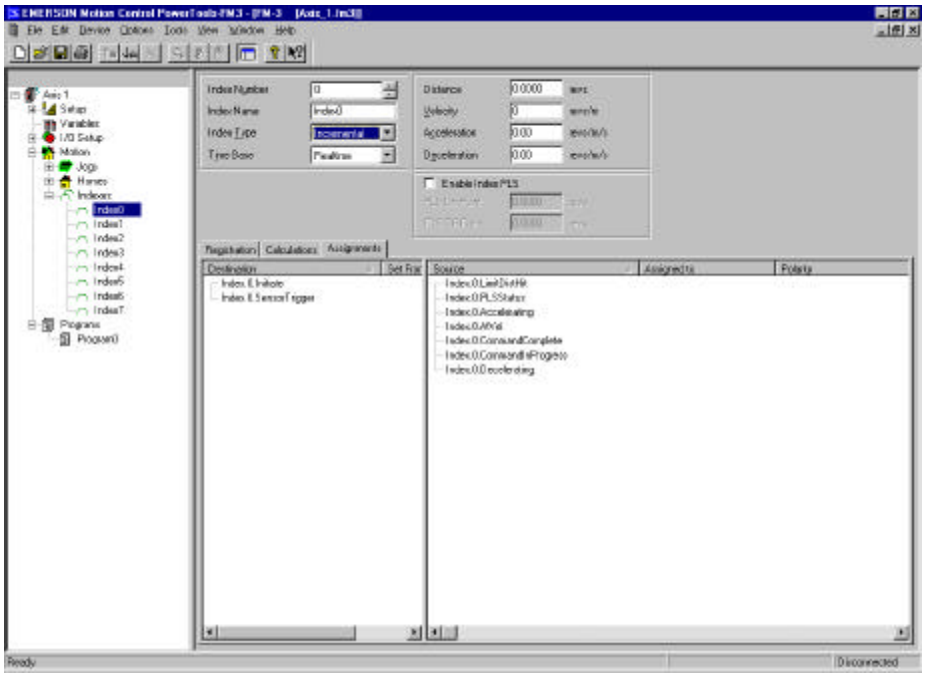


Figure 75: Index Setup View (Incremental Type)

Now that all of the motion parameters are setup, the jogs and indexes can be initiated through assignments to hardware inputs that were made on the assignment view. See Figure 16 to determine which input is used to activate which functions.

This example specifies that the indexes must be initiated from assignments, as well as from a program. So a program must be created using Step 4 of this guide.

Upon initiating the program, it will wait until DriveInput.1 is activated, then run all three indexes with a 1 second dwell in between each, and then loop back to the top of the program. The program will run continuously, or until the Stop destination is activated.

The example program uses the following instructions:

Do While / Loop – This program instruction is used to repeat a sequence of code as long as a test expression is true. The test expression is validated before the loop is entered. If the test expression is evaluated as False (0), then the code inside the loop will be skipped over. If the expression is evaluated as True (1), then the code inside the loop will be performed. Upon reaching the Loop instruction, the program flow returns to the Do While instruction, and the expression is evaluated again. The example program uses an expression of TRUE, which is always evaluated as True (1). Therefore the program will run continuously.

Wait For – This program instruction halts program execution until the expression becomes True. Once True, the program continues with the next line of code. The example program uses a Wait For DriveInput.1 so that the program will wait until DriveInput.1 is activated, then continue on.

Index.Initiate – This program instruction is used to initiate a single index. When using this instruction, an Index Selection Box will appear allowing you to select the index you wish to initiate. This selection box is shown in Figure 21.

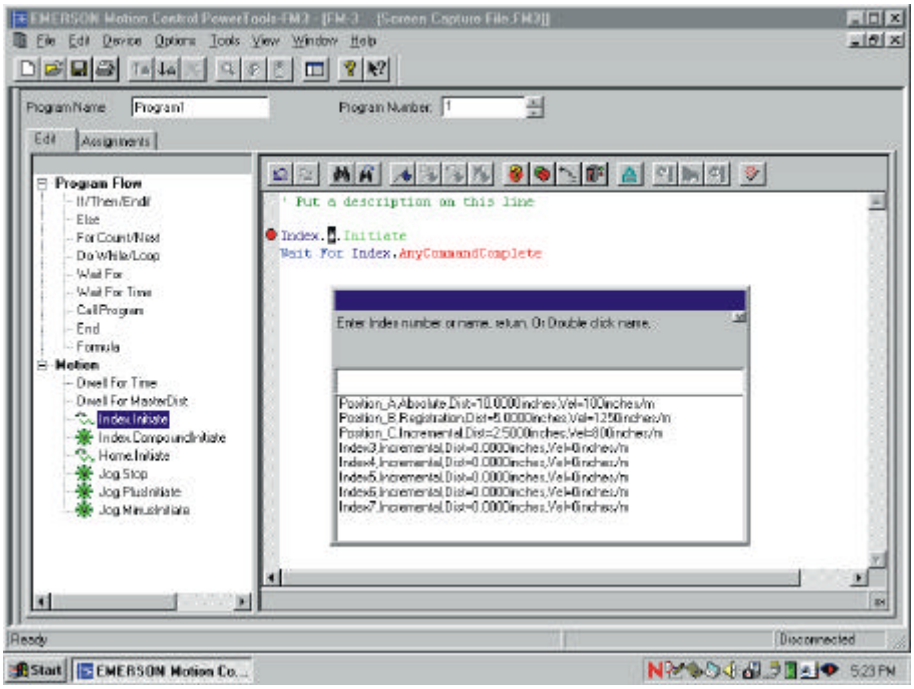


Figure 76: Index Selection Box

The program is initiated using DriveInput.4 (see Figure 17).

The setup is now complete, and the example is ready to be downloaded.

Tuning Procedures

The drive uses closed loop controllers to control the position and velocity Travel Limit of the attached motor. These position and velocity controllers and the associated tuning parameters are in effect when the drive is in velocity or pulse mode and have no effect when the drive is in Torque mode.

Many closed loop controllers require tuning using individual user-specified proportional, integral and derivative (PID) gains which require skilled “tweaking” to optimize. The combination of these gains along with the drive gain, motor gain, and motor inertia, define the system bandwidth. The overall system bandwidth is usually unknown at the end of the tweaking process. The drive closes the control loops for the user using a state-space pole placement technique. Using this method, the drive’s position control can be simply and accurately tuned. The overall system’s bandwidth can be defined by a single user-specified value (Response).

The drive’s default settings are designed to work in applications with up to a 10:1 load to motor inertia mismatch. Most applications can operate with this default setting.

Some applications may have performance requirements which are not attainable with the factory settings. For these applications a set of measurable parameters can be specified which will set up the internal control functions to optimize the drive performance. The parameters include Inertia Ratio, Friction, Response and Line Voltage. All the values needed for optimization are “real world” values that can be determined by calculation or some method of dynamic measurement.

PID vs. State-Space

The power of the state-space control algorithm is that there is no guessing and no “fine tuning” as needed with user-specified PID methods. PID methods work well in controlled situations but tend to be difficult to setup in applications where all the effects of the system are not compensated for in the PID loop. The results are that the system response is compromised to avoid instability.

The drive state-space control algorithm uses a number of internally calculated gains that represent the wide variety of effects present in a servo system. This method gives a more accurate representation of the system and maximizes the performance by minimizing the compromises.

You need only to setup the system and enter three parameters to describe the load and the application needs. Once the entries are made the tuning is complete - no guessing and no “tweaking”. The drive uses these entries plus motor and amplifier information to setup the internal digital gain values. These values are used in the control loops to accurately set up a stable, repeatable and highly responsive system.

Tuning Procedure

Once the initial setup has been completed, you can run the system to determine if the level of tuning is adequate for the application. There are basically four levels of tuning for a drive.

- No Tuning
- Basic Level
- Intermediate Level
- Fully Optimized Level

Each level is slightly more involved than the previous one requiring you to enter more information. If your system needs optimization, we recommend that you start with the Basic Level, then determine if further tuning is needed based on axis performance.

The setup procedures explained here assume that you are using EMERSON Motion Control PowerTools® FM-3 software or an FM-P.

Initial settings

Set the drive tuning parameters as follows:

- Inertia Ratio = 0
- Friction = 0
- Response = 50
- High Performance Gains = Enabled
- Feedforwards = Disabled

Tuning steps

If your Inertia Ratio is greater than 10 times the motor inertia go directly to the Intermediate Level tuning.

No Tuning

No tuning will be required in most applications where the load inertia is 10 times the motor inertia or less.

Basic Level

Adjust Response to obtain the best performance.

Intermediate Level

1. Calculate or estimate the load inertia. It is always better to estimate low.
2. Disable the drive.

3. Enter the inertia value calculated into the Inertia Ratio parameter.
4. Set the Line Voltage to the applied voltage (default is 230 VAC).
5. Leave all other tuning parameters at the initial values.
6. Enable the drive and run the system.
7. Adjust Response to obtain the best performance.

Fully Optimized Level

1. Determine the actual system parameters.
2. Disable the drive.
3. Enter the parameters.
4. Line Voltage set to the applied voltage (default is 230 VAC).
5. Enable the drive and run the system.
6. Adjust Response to obtain the best performance.

General Tuning Hints

The Response is normally the final adjustment when tuning. For best performance the Response should be lower with a higher inertia mismatch (>10:1) and higher with a lower inertia mismatch.

If your system has some torsional compliance, such as with a jaw type coupling with a rubber spider, or if there is a long drive shaft, the Response should be decreased. The highest recommended Response with High Performance Gains enabled is 100 Hz.

Also, enabling the Low Pass Filter helps diminsh the resonant frequency of torsionally compliant loads. In such cases, using the Low Pass Filter usually allows for higher Response values. The optimum Low Pass Filter frequency is at the frequency of the resonance.

Feedforwards can be enabled if the performance requirements are very demanding. However when using them, make sure the Inertia Ratio and Friction values are an accurate representation of the load. Otherwise, the system performance will actually be degraded or stability will suffer. Enabling the Feedforward makes the system less tolerant of inertia or friction variations during operation.

Tuning Parameters

Inertia Ratio

Inertia Ratio specifies the load to motor inertia ratio and has a range of 0.0 to 50.0. A value of 1.0 specifies that load inertia equals the motor inertia (1:1 load to motor inertia). The drives can control up to a 10:1 inertia mismatch with the default Inertia Ratio value of 0.0. Inertial mismatches of over 50:1 are possible if response is reduced.

The Inertia Ratio value is used to set the internal gains in the velocity and position loops, including feedforward compensation if enabled.

To calculate the Inertia Ratio value, divide the load inertia reflected to the motor by the motor inertia of the motor. Include the motor brake as a load where applicable. The resulting value should be entered as the Inertia Ratio parameter.

$$IR = \frac{RLI}{MRI}$$

Where:

IR = Inertia Ratio
RLI = Reflected Load Inertia (lb-in-sec^2)
MI = Motor Inertia (lb-in- sec^2)

If the exact inertia is unknown, a conservative approximate value should be used. If you enter an inertia value higher than the actual inertia, the resultant motor response will tend to be more oscillatory.

If you enter an inertia value lower than the actual inertia, but is between 10 and 90 percent of the actual, the drive will tend to be more sluggish than optimum but will usually operate satisfactorily. If the value you enter is less than 10 percent of the actual inertia, the drive will have a low frequency oscillation at speed.

Friction

In the drive, this is a viscuous friction parameter, characterized in terms of the rate of friction increase per 100 motor RPM. The range is 0.00 to 100.00 in units of percent continuous torque of the specified motor/drive combination. The Friction value can either be estimated or measured.

If estimated, always use a conservative (less than or equal to actual) estimate. If the friction is completely unknown, a value of zero should be used. A typical value used here would be less than one percent.

If the value entered is higher than the actual, system oscillation is likely. If the value entered is lower than the actual a more sluggish response is likely but generally results in good operation.

Response

The Response adjusts the velocity and position loop bandwidths with a range of 1 to 500 Hz. In general, it affects how quickly the drive will respond to commands, load disturbances and velocity corrections. The effect of Response is greatly influenced by the status of the High Performance Gains.

With High Performance Gains disabled, the actual command bandwidth of the drive system will be equal to the Response value. In this case the load disturbance correction bandwidth is very low (approximately 0.1 Hz). Increasing the Response value will reduce the drive's response time to velocity command changes but will not affect the response to load or speed disturbances.

When High Performance Gains are enabled, the Response bandwidth is set to the Response value. In this case, it reflects both the velocity command and the load disturbance correction bandwidth. Increasing the Response when the High Performance Gains are enabled will increase loop stiffness. With High Performance gains enabled, the maximum Response level recommended is approximately 100 Hz.

If the Inertia Ratio and Friction values are exactly correct and the High Performance Gains are enabled, changing the Response will not affect the damping (percent of overshoot and number of ringout cycles) to velocity command changes or load disturbance corrections but will affect their cycle frequency. The response level generally should be decreased as the load to motor inertia ratio increases or if High Performance Gains are enabled.

Feedforwards

Feedforward gains are essentially open loop gains that generate torque commands based on the commanded velocity, accel/decel and the known load parameters (Inertia Ratio and Friction). Using the feedforwards reduces velocity error during steady state and reduces overshoot during ramping. This is because the Feedforwards do not wait for error to build up to generate current commands.

Feedforwards should be disabled unless the absolute maximum performance is required from the system. Using them reduces the forgiveness of the servo loop and can create instability if the actual inertia and/or friction of the machine varies greatly during operation or if the Inertia Ratio or Friction parameters are not correct.

The internal feedforward velocity and acceleration gains are calculated by using the Inertia Ratio and Friction parameters. The feedforward acceleration gain is calculated from the Inertia Ratio parameter and the feedforward velocity gain is calculated from the Friction parameter.

When Feedforwards are enabled, the accuracy of the Inertia Ratio and Friction parameters is very important. If the Inertia Ratio parameter is larger than the actual inertia, the result would be a significant velocity overshoot during ramping. If the Inertia parameter is smaller than the

actual inertia, velocity error during ramping will be reduced but not eliminated. If the Friction parameter is greater than the actual friction, it may result in velocity error or instability. If the Friction parameter is less than the actual friction, velocity error will be reduced by not eliminated.

Feedforwards can be enabled in any operating mode, however, there are certain modes in which they do not function. These modes are described in table below.

Operating Mode	Feedforward Parameters Active	
	Accel FF	Vel FF
Analog Velocity	No	Yes
Preset Velocity	Yes	Yes
Pulse/Position	No	No
Summation	No	Yes

Low Pass Filter Group

The Low Pass Filter will reduce machine resonance due to mechanical coupling and other flexible drive/load components by filtering the command generated by the velocity loop. A check box on the Tuning tab enables a low pass filter applied to the output of the velocity command before the torque compensator. The low pass filter frequency parameter defines the low pass filter cut-off frequency. Signals exceeding this frequency will be filtered at a rate of 40 dB per decade. The default value is 600Hz.

Line Voltage

Line Voltage specifies the applied power and adjusts the internal gains to compensate for it. This parameter has two choices 115 VAC and 230 VAC. If the Line Voltage is set to 230 VAC when the actual applied voltage is 115 VAC, the motor will be slightly less responsive to commands and load disturbances.

Warning

The Line Voltage must never be set to 115 VAC if the applied voltage is actually 230 VAC. This can cause drive instability and failure.

Determining Tuning Parameter Values

For optimum performance you will need to enter the actual system parameters into the drive. This section discusses the methods which will most accurately determine those parameters.

Note

If you have an application which exerts a constant unidirectional loading throughout the travel such as in a vertical axis, the inertia tests must be performed in both directions to cancel out the unidirectional loading effect.

Initial Test Settings

When running the tests outlined in this section, the motor and drive must be operational so you will need to enter starting values.

If your application has less than a 10:1 inertia mismatch, the default parameter settings will be acceptable. If the inertial mismatch is greater than 10:1, use the following table for initial parameter settings.

Parameter	Setting
Friction	0.00
Inertia Ratio	1/3 to 1/2 Actual
Response	500/Inertia Ratio
High Performance Gains	Disabled
Feedforwards	Enabled
Line voltage	Actual Applied

Determining Friction

This parameter represents friction that increases proportionally as motor velocity increases. The viscous friction of your system can be determined by reading the percent of continuous torque required to operate the loaded motor at two different speeds.

Consider the following before determining the Friction:

- The most consistent readings can usually be obtained at motor speeds higher than 500 RPM but lower test speeds can be used if necessary.
- If your application has travel limits, it may be helpful to use an external position controller to prevent the axis from exceeding the machine limits. Set up a trapezoidal profile as shown.
- In the procedure below, the Torque Command and Velocity Feedback parameters can be measured using the drive's analog outputs, PowerTools software or an FM-P.
- With vertical loads the test readings must be taken while traveling in the same direction.
- An oscilloscope may be needed for systems with limited travel moves to measure the rapidly changing torque and velocity signals.
- If your system's friction changes with operating temperature, perform this procedure at normal operating temperature.

Procedure for Determining Friction:

1. Run the motor at the low test speed (at least 500 RPM).
2. While at speed, note the Torque Command Actual value (TCL).

Note

If the friction loading of your system varies when operating at constant speed, due to a load or spring load that changes as the motor rotates, use the lowest value measured.

3. Repeat Step 1 using a velocity at least two times the low speed.

4. While at speed, note the Torque Command Actual value (TCH).

5. Use the following formula to calculate the friction:

$$FV = (100)\frac{(T_{CH} - T_{CL})}{RPM_H - RPM_L}$$

Where:

T_{CH} = Torque Command Limited value at higher speed
T_{CL} = Torque Command Limited at lower speed
RPM_H = Higher RPM (velocity)
RPM_L = Lower RPM (velocity)
FV = Friction value

The figure below shows the relationship of Torque Command to the Velocity Feedback. There is increased torque during the Accel ramp (Ta), constant torque (Tc) during the constant velocity portion of the ramp and decreased torque (Td) during the decel ramp.

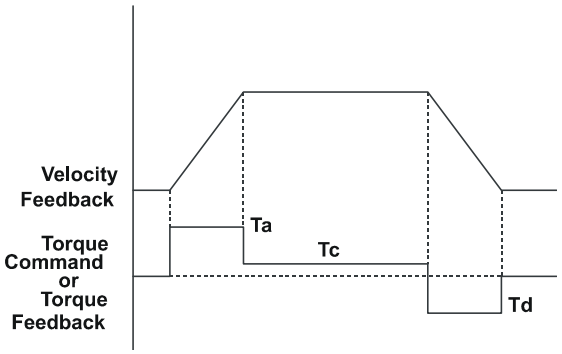


Figure 77: Trapezoidal Velocity Waveform with Torque Waveform

Determining Inertia Ratio

Actual system Inertia Ratio is determined by accelerating and decelerating the load with a known ramp while measuring the torque required.

Consider the following before determining the inertia:

- If your application allows a great deal of motor motion without interference, it is recommended that you use a Preset Velocity to produce accurate acceleration ramps.

- If your application has a very limited range of motion, it is recommended that you use a position controller to produce the acceleration ramps and to prevent exceeding the axis range of motion.
- The accel and decel ramp should be aggressive enough to require at least 20 percent of continuous motor torque. The higher the torque used during the ramp, the more accurate the final result will be.
- With ramps that take less than 1/2 second to accelerate, read the Diagnostic Analog Outputs with an oscilloscope to measure the Torque Feedback.
- With ramps that take 1/2 second or longer to accelerate, read the Torque Command parameter on the Motor tab of PowerTools or with the Watch Window.
- To best determine the inertia, use both acceleration and deceleration torque values. The difference allows you to drop the constant friction out of the final calculation.
- If your application exerts a constant "unidirectional loading" throughout the travel such as in a vertical axis, the inertia test profiles must be performed in both directions to cancel out the unidirectional loading effect.
- The Torque Command Limited and Velocity Feedback parameters can be measured using the drive's Analog Outputs, PowerTools software or an FM-P.

An oscilloscope will be needed for systems with limited travel moves and rapidly changing signals of torque and velocity.

Inertia Measurement Procedure:

Note

The test profile will need to be run a number of times in order to get a good sample of data.

1. Enable the drives and run the test profiles.
2. Note the Torque Command Limited value during acceleration and deceleration.
3. Use the appropriate formula below to calculate the inertia.

For horizontal loads or counterbalanced vertical loads use the following formula:

$$IR = 1 - \frac{(R \bullet Vm\,(Ta + Td))}{2000}$$

Where:

IR = Inertia Ratio
R = ramp in ms/kRPM
Ta = (unsigned) percent continuous torque required during acceleration ramping (0 - 300)

Td = (unsigned) percent continuous torque required during deceleration ramping (0 - 300)
Vm = motor constant value from Table 18 below

For un-counter balanced vertical loads use the following formula:

$$IR = 1 - \frac{(R \bullet Vm\,(Tau + Tud + Tad + Tdd))}{4000}$$

Where:

IR = Inertia Ratio
R = ramp in ms/kRPM
Vm = motor constant value from the table below
Tau = (unsigned) percent continuous torque required during acceleration ramping while moving up (against the constant force)
Tdu = (unsigned) percent continuous torque required during deceleration ramping while moving up (against the constant force)
Tad = (unsigned) percent continuous torque required during acceleration ramping while moving down (aided by the constant force)
Tdd = (unsigned) percent continuous torque required during deceleration ramping while moving down (aided by the constant force)

Ramp Units Conversion

If you are using an external position controller to generate motion you may need to connect the ramp units as desired below.

Many position controllers define acceleration in units per sec^2. The formulas above use ms/kRPM. Make sure you make this conversion when entering the information into the formula.

Conversion Formula:

$$MPK = \frac{10^6}{(RPSS \bullet 60)}$$

Where:

MPK = accel ramp in ms/kRPM
RPSS = accel ramp in revolutions per $second^2$

Motor	Drive	Vm	Percent Continuous/volt Scaled Torque Command Output (default))	RPM /volt Scaled Velocity Command Output (default))
MG-205	EN-204	4.77	30	600
MG-208		5.11	30	600
MG-316		3.17	30	600
MG-316	EN-208	3.17	30	600
MG-340		3.14	30	600
MG-455		2.44	30	600
MG-455	EN-214	2.44	30	600
MG-490		1.85	30	600
MG-4120		1.69	30	600

Diagnostics and Troubleshooting

Diagnostic Display

The diagnostic segment display on the front of the drive shows drive status, FM-3 status, and fault codes. When a fault condition occurs, the drive will display the fault code, overriding the status code. The alphanumeric display on the FM-3 provides additional fault information.

The decimal point is “On” when the drive is enabled, and the stop input is not active. This indicates that the drive is ready to run and any motion command will cause motion. Motion commands will not cause motion unless you are Ready (R) and the decimal point is “On”.

Display Indication	Status	Description	FM-3 Keypad Display
	Ready	The FM-3 and E Series drive system is functioning normally and is ready to execute a motion command.	Drive Type Address Motor Type Menu Groups
	Disabled	Power Stage is disabled.	Drive Type Address Motor Type Menu Groups
	RMS Foldback	Motor torque is limited to 80 percent.	Drive Type Address Motor Type Menu Groups
	Stall Foldback (E Series drive only)	Drive output current is limited to 80 percent of drive stall current.	Drive Type Address Motor Type Menu Groups
	Ready to Run	Drive enabled. No stop input.	Drive Type Address Motor Type Menu Groups
	Homing	Home is executing.	Drive Type Address Motor Type Menu Groups
	Indexing	Index is executing.	Drive Type Address Motor Type Menu Groups

Display Indication	Status	Description	FM-3 Keypad Display
	Jogging	Jog is executing.	Drive Type Address Motor Type Menu Groups
	Program	Program is executing.	Drive Type Address Motor Type Menu Groups

Fault Codes

A number of diagnostic and fault detection circuits are incorporated to protect the drive. Some faults, such as high DC bus and amplifier or motor over temperature can be reset with the Reset button on the front of the drive or the Reset input function. Other faults, such as encoder faults, can only be reset by cycling power “Off” (wait until the diagnostics display turns “Off”), then power “On”.

The drive accurately tracks motor position during fault conditions. For example, if there is a "Low DC Bus" fault where the power stage is disabled, the drive will continue to track the motor’s position provided the logic power is not interrupted.

The +/- Limit faults are automatically cleared when the fault condition is removed. The table below lists all the fault codes in priority order from highest to lowest. This means that if two faults are active, only the higher priority fault will be displayed.

Segment Display	Fault	Action to Reset	Bridge Disabled	FM-3 Display
	Flash Invalid	Reprogram the FM’s Flash	Yes	
	Watchdog Timer	Button or Input	Yes	Watch Dog
	Power Up Test	Power	Yes	Power Up Tst
	NVM Invalid	Button or Input	Yes	NVM Invalid

Segment Display	Fault	Action to Reset	Bridge Disabled	FM-3 Display
	Invalid Configuration	Button or Input	Yes	Invalid Conf
	Power Module	Button or Input	Yes	Power Module
	High DC Bus	Button or Input	Yes	High DC Bus
	Low DC Bus	Button or Input	Yes	Low DCBus
	Encoder State	Power	Yes	Encoder State
	Encoder Hardware	Power	Yes	Encoder HW
	Motor Overtemp	Button or Input	Yes	MotorOvrTemp
	RMS Shunt Power	Button or Input	Yes	RMS ShuntPowr
	Overspeed	Button or Input	Yes	Over Speed
	Max Following Error	Button or Input	Yes	Following Err
	Travel Limit +/-	Auto	No	TravLmtMinus TravLmtPlus

Segment Display	Fault	Action to Reset	Bridge Disabled	FM-3 Display
	All "On"	Normally on for one second during power up	Yes	
	FM-3			** Fault ** ***** Depend on Fault

Fault Descriptions

Flash Invalid

This fault indicates that the firmware checksum has failed. Use the Tools Program Flash menu item from Emerson Motion Control PowerTools® to reprogram/upgrade the firmware stored in flash memory. If this problem persists, call EMERSON Motion Control. A common cause would be an interrupted F/W Flash upgrade (cable disconnected during an upgrade process).

Watchdog Timer

The watchdog timer ensures that the firmware is operating normally. If the firmware fails to reset this timer every 10 ms, the drive hardware will be reset, all inputs and outputs will be cleared (“Off”), and a Watchdog Timer fault is generated.

Power Up Test

This fault indicates that the power-up self-test has failed. This fault cannot be reset with the reset command or reset button.

NVM Invalid

At power-up the drive tests the integrity of the non-volatile memory. This fault is generated if the contents of the non-volatile memory are invalid.

FM-3 Fault

A 3 will be displayed in the segment display on the E Series drive when the FM-3 experiences a fault. See the table above for more information about the various faults.

U — Invalid Configuration

The FM was not on this drive during its previous power-up and it is not known if the setup data in the FM matches the drive and motor to which the FM is now attached.

This can also happen when a FM is removed from a drive and the drive is powered-up.

To reset the fault, create or open a configuration file with the correct drive and motor selections and download the configuration to the FM or drive. If you are certain that the setup data in the FM or drive matches the system configuration, press and hold the E Series drive's Reset button for 10 seconds (until the fault is cleared).

Caution

Damage may occur to the drive, motor or both if the fault is cleared using the Reset button when the setup data in the FM does not match the current drive and motor.

Z — Power Module

This fault is generated when a power stage over-temperature, over-current or loss of power stage logic supply occurs. This can be the result of a motor short to ground, a short in the motor windings, a motor cable short or the failure of a switching transistor.

It can also occur if the drive enable input is cycled "Off" and "On" rapidly (>10 Hz).

H — High DC Bus

This fault will occur whenever the voltage on the DC bus exceeds 440 VDC. The most likely cause of this fault would be an open shunt fuse, a high AC line condition or an application that requires an external shunt (that is, a large load with rapid deceleration).

L — Low DC Bus

This fault will occur whenever the voltage on the DC bus drops below 96 volts. The most likely cause of this fault is a reduction (or loss) of AC power. A 50 ms debounce time is used with this fault to avoid faults caused by intermittent power disruption.

E — Encoder State

Certain encoder states and state transitions are invalid and will cause the drive to report an encoder state fault. This is usually the result of noisy encoder feedback caused by poor shielding.

E — Encoder Hardware

If any pair of encoder lines are in the same state, an encoder line fault is generated. The most likely cause is a missing or bad encoder connection.

M — Motor Overtemp

This fault is generated when the motor thermal switch is open due to motor over-temperature or incorrect wiring.

RMS Shunt Power

This fault is generated when RMS shunt power dissipation is greater than the design rating of the internal shunt.

Overspeed

This fault occurs when the actual motor speed exceeds the Overspeed Velocity Limit parameter. This parameter can be accessed with PowerTools FM-3 software.

F — Max Following Error

This fault is generated when the following error exceeds the following error limit (default following error limit is .2 revs). With PowerTools FM-3 you can change the Following Error Limit value on disable in the Position view.

L — Travel Limit +/-

This fault is caused when either the + or - Travel Limit input function is active.

All "On"

This is a normal condition during power up of the drive. It will last for less than 1 second. If this display persists, call EMERSON for service advice.

FM-3 Specific Fault Descriptions

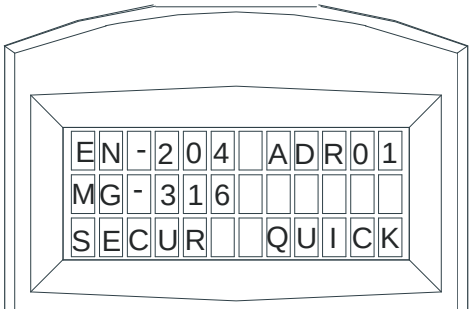


Figure 78: FM-3 display is used to display drive and FM-3 faults

Traj Fault

This fault occurs when the drive has received trajectory data from the FM-3 that indicates a problem. Check the user units, velocities, accels and decels for correct values.

ISR Overrun

This fault is generated when a module flash memory problem occurs. To correct, replace the FM-3.

OutofSync

This fault is generated when the system is out of sync. To correct it, press the reset button and check the configuration.

ProgramFault

This fault indicates an FM-3 user program fault. For example the program is attempting to divide by zero, or overflows and math errors caused by numbers that are too large, or non-existing parameters.

ProgramInvalid

The user program in flash memory will not run. Download the User Program again using PowerTools FM-3. A common cause of this problem could be an interrupted configuration download, such as a cable being disconnected during a download.

Brake Operation

MG motor brake operation is controlled by the Brake Release and Brake Control destinations. These destinations can be used together to control the state of the Brake source. The table below shows the relationship between the Brake sources and destinations (see “Diagnostic Display”).

Note

No motion should be commanded while the brake is engaged.

Brake Release Destination		Off		On	
Brake Control Destination		On	Off	On	Off
Drive Power Stage	Enabled	0	1	1	1
	Disabled	0	0	1	1

Brake Release

The Brake Release destination function will release the brake under all conditions. When this function is active, the Brake output will be on (that is, release brake). This function overrides all other brake control, thus allowing the brake to be released while a fault is active or the power stage is disabled. See also Brake source function.

Brake Control

The Brake destination function, when active, will engage the brake unless overridden by the Brake Release function. This function lets you externally engage the brake while allowing the drive to also control the brake during fault and disabled conditions.

Brake

The Brake source function is used to control the motor holding brake. If the Brake function is off, the brake is mechanically engaged. When the brake is engaged, the diagnostic display on the front of the drive will display a “b”.

The drive outputs are limited to 150 mA capacity, therefore, a suppressed relay is required to control motor coil. EMERSON Motion Control offers a relay, model # BRM-1.

Analog Outputs

The drive has two 10 bit Analog Outputs which may be used for diagnostics, monitoring or control purposes. These outputs are referred to as Channel 1 and Channel 2. They can be accessed from the command connector on the drive or from the diagnostics output pins located on the front of the drive.

Each Channel provides a programmable Analog Output Source. (In early releases, the FM-3 analog output is fixed at the default selections.)

Default Analog Output Source:

- Channel 1 = Velocity Feedback
- Channel 2 = Torque Command

Output	Source	Offset	Scale
1	Velocity Feedback	0	600 RPM/volt
2	Torque Command	0	30% /volt

Each channel includes a programmable Analog Output Offset and an Analog Output Scale. This feature allows you to “zoom in” to a desired range effectively increasing the resolution. The units for both of these parameters is dependent upon the Analog Output Source selection.

Analog Output Offset units:

- Velocity Feedback = RPM
- Torque Command = Percent of continuous torque

Analog Output Scale units:

- Velocity Feedback = RPM/volt
- Torque Command = Percent of continuous torque/volt

Example:

You could use the Analog Outputs to accurately measure velocity overshoot. For example, to measure a target velocity of 2000 RPM at a resolution of ±10V = ±200 RPM do the following.

1. Selected Velocity Feedback for the Analog Output Source.
2. Set the Analog Output Offset to 2000 RPM.
3. Set the Analog Output Scale to 20 RPM/VOLT.

This will provide an active range from -10 to +10 Volts to represent 1800 to 2200 RPM. Therefore, the measured resolution has been increased.

Diagnostic Analog Output Test Points

The DGNE cable was designed to be use with either an oscilloscope or a meter. The wires are different lengths to avoid shorting to each other. However, if signals do get shorted to GND, the drive will not be damaged because the circuitry is protected.

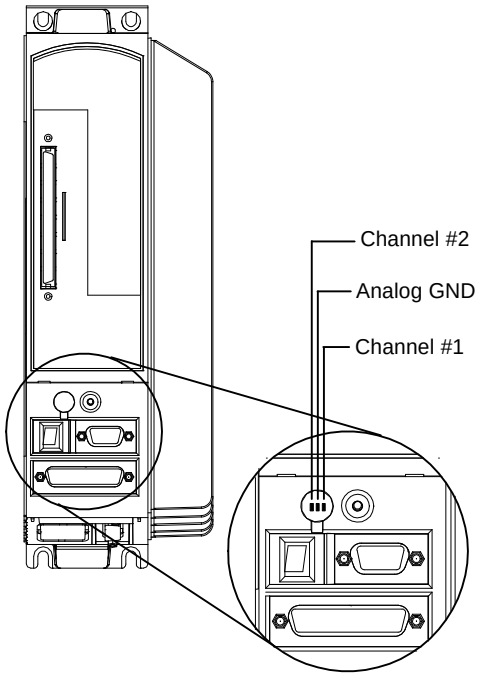


Figure 79: Diagnostic Output Test Points

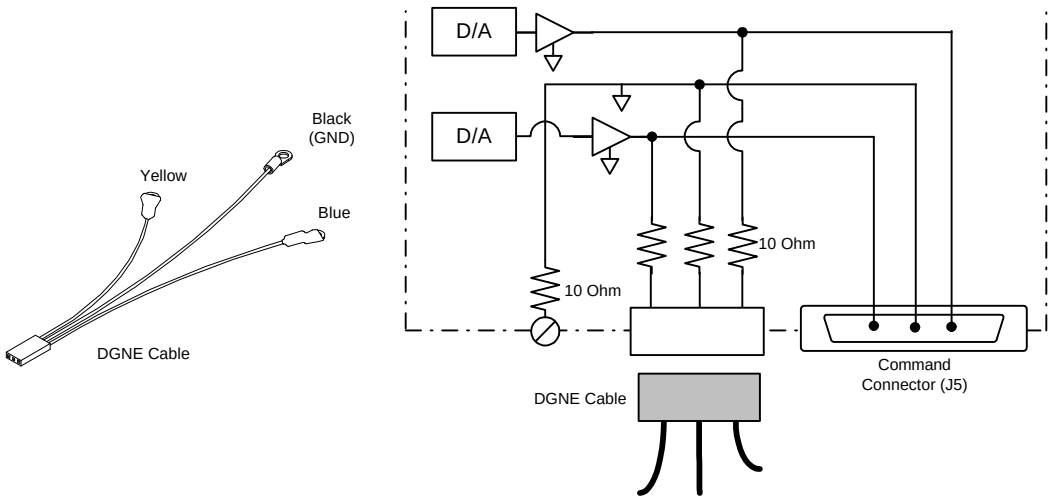


Figure 80: Diagnostic Cable (DGNE) Diagram

Drive Faults

The Active Drive Faults dialog box is automatically displayed whenever a fault occurs. There are two options in this dialog box: Reset Faults and Ignore Faults.

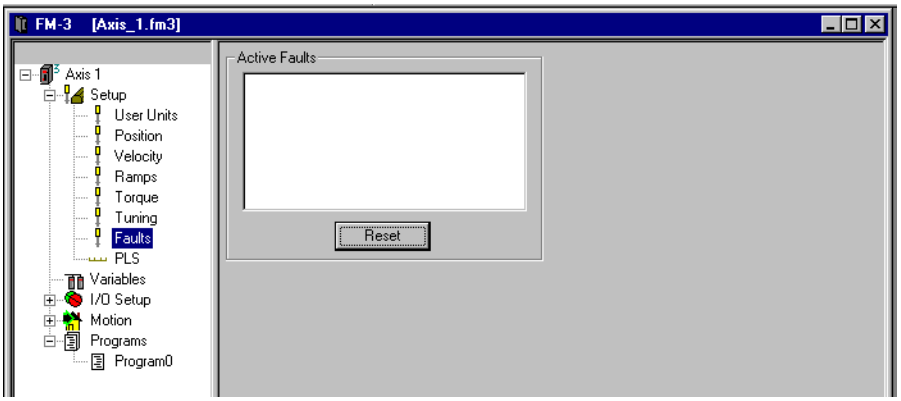


Figure 81: Active Drive Faults Dialog Box

Resetting Faults

Some drive faults are automatically reset when the fault condition is cleared. Others require drive power to be cycled or the drive to be "rebooted" to be cleared. If you wish to continue working in the PowerTools software without resetting the fault, click the *Ignore Fault* button.

To reset faults that can be reset with the *Reset Faults* button, simply click the *Reset Faults* button in the Drive Faults Detected dialog box or push the Reset button on the front of the drive where the fault occurred.

Viewing Active Drive Faults

To view all active drive faults, select the View Faults command from the **Device** menu. The dialog box displayed is the same as Active Drive Faults dialog box described above.

Rebooting the Drive

To reboot the drive, cycle power or select the Reboot Drive command from the **Device** menu. This command reboots the drive attached to the active Configuration Window.

Error Messages

PowerTools will pop-up an error message box to alert you to any errors it encounters. These message boxes will describe the error and offer a possible solution.

The terms below appear in the list the of common problems you might encounter when working with PowerTools software along with the error message displayed, the most likely cause and solution.

Assign means to set a value using an equation. For example, x = 2, you are assigning the value of 2 to x.

A **Boolean value** is a value that represents two states such as On or Off. In the FM3 there are three variable types that have Boolean values. They are Boolean Variables, Input Event Variables and Output Event Variables. They all have a Boolean Value and can be used in equations to assign their Boolean value to another variable or in a conditional test.

In an equation, conditional tests such as (vel > 3.1) become Boolean values.

In FM3 Programming, unquoted text names are used to represent Boolean constants. Several different names are available. They all represent the two Boolean states and therefore are interchangeable.

An **Expression** is a collection of mathematical operands (variables, constants and numbers) and operators(+, -, *, <, >, etc) that form a value. The right hand side of an equation (to the right of the =) is an expression.

The **Parser** is an internal component of PowerTools software that reads your program text file and generates executable code used by the FM3 Module firmware. The parser detects errors that are reported to you as Red Dot Error Messages.

Your program errors are displayed in the program view in **Red Dot Error Messages**. They are indicated with red dots. To get further information on the cause of the error you use the programming tool bar's red dotted question mark. This is an on/off setting that enables error message displays and application help messages.

Non-Programming Error Messages

These messages occur while you are working in a view other than the Program View. The Program View has error messages specific to it, and they are described in a Programming Error Messages section. The Popup messages are listed below.

Can Not Add Index, until current is valid.,
Can Not Add Jog, until current is valid.,
Can Not Add Program, until current is valid.

The current View must be valid before you can create a new instance of Index, Jog or Home.

Error: The maximum limit of Instances is reached.

The number of Index instances, Jog Instances and Home Instances is limited. If you attempt to add an instance and the number of existing instances is at the maximum, you will get this message.

Programming Error Messages

These Red Dot Error messages occur while you are working in the Program View.

When creating a program, the parser is executed when you left mouse click, when you arrow off the current Line, when you enter the carriage return, when you paste or when you drop a drag source. The parser detects errors and marks the line with a “Red Dot”. To get further information on the cause of the error you use the programming tool bar’s red dotted question mark. This is an on/off setting that enables error message displays and application help messages.

Error: Your Application is not valid to download ...

This message lets the user know that there are errors such as "Red Dot" errors in one or more programs that prevent the program from being downloaded to the module. The message will provide more information such as which program is invalid to help the user correct the problem.

FYI: Your Application has ...

This message lets the user know that there are errors such as "Red Dot" errors in one or more programs. The operation (i.e. file save) was completed, however other operations such as download would fail for this application. The message will provide more information such as which program is invalid to help the user correct the problem.

Error: A FM3 number’s decimal Point resolution can not be greater than ten

The FM3 does not use standard floating point. It uses Integer arithmetic to prevent round off errors. Decimals are used, but decimal point position is handled separately from the integer value. Zero puts the decimal point to the far right. Ten puts the decimal point at the far left. The mantissa must be between -2147483648 and 214748364. The decimal point position must be between zero and ten.

Error: A FM3 number’s mantissa must be between -2147483647 and 2147483647.

The mantissa must be between -2147483648 and 214748364. The decimal point position must be between zero and ten.

Error: A numeric variable can only be assigned a numeric value

The Variable is a numeric. It only accepts types consisting of numeric values.

Error: A string variable can only be assigned a quoted text string

The Variable is string. It only accepts types consisting of text strings.

Error: Can only compare(>,<,etc) numeric results

This message occurs in conditional Expressions (i.e. If then). Variables are type identified, so equation and assignments (x = 9) can be verified. In an expression only numerical values can be compared for greater than and less than conditions.

Error: Couldn't find named Variable

Could not find the variable.

Error: Couldn't find named Variable from Text

Could not find the variable defined by program text.

Error: Destination Event variables can only be assigned an Event or Boolean

The Variable is an Input event. It only accepts types consisting of Boolean, and events.

Error: Program Instance does not Exist

You attempted the “Call Program.#”, but the program does not exist.

Error: Single value expressions can only be Boolean constants, Events or Boolean variables

This message occurs in conditional Expressions (i.e. If then). Variables are type identified, so equation and assignments .(x = 9) can be verified. In an expression you can use single variables without a comparison, but , then they must be a Boolean constants, Events or Boolean variable.

Error: String does not represent a predefined name..

The string needs to match one of the defined strings on record in the FM3 Registry data base.

Error: String is not a selection.

The string matches one of the defined strings, but that string is not a selection.

Error: Syntax error encountered

Parser Error Message. The Parser can not understand your text sequence.

Error: Text Strings are limited to 12 characters...

To change a Name you assign a quoted text string to that name. In FM3, text strings are fixed at 12 characters. If you use fewer than 12 characters, blanks are automatically inserted. An error occurs if you attempt to use more than 12 characters.

Error: The Boolean variables can only be assigned an Event or Boolean value.

The Variable only accepts types consisting of Boolean, and events.

Error: The destination variable does not accept Data

The Variable's internal data type attribute was not found.

Error: The destination variable does not accept negative numbers

You attempted to assign a negative number to an unassigned variable .

Error: The destination variable is Read Only

This message occurs when trying to assign a value to a read only variable.

Error: The destination variable only accepts a numeric value

Error: The destination variable only accepts a Boolean or Event value

Error: The destination variable only accepts quoted "text"

Error: The destination variable only accepts selection text

Error: The destination variable's resolution is less than the resolution of the number

You attempted to assign a number with a greater resolution of decimal points than the variable will accept (ie index.0.vel = 2.34567). The User Units setup will allow you to define the desired decimal point resolution.

Error: The number is outside the range of the destination variable

You attempted to assign a number that is outside the variable's range. To determine the range comment out this instruction and use the red dot help on the variable.

Error: The mix of variable or expressions types can not be added or subtracted

This message occurs in equations. Variables are type identified, so equation and assignments (x = 9) can be verified. In an equation only numerical values can be multiplied or divided. Booleans, Selections, text and events can not be added.

Error: The mix of variable or expressions types can not be compared(=)

This message occurs in conditional Expressions (i.e. If then). Variables are type identified, so equation and assignments (x = 9) can be verified. In an expression numerical, Boolean and event values can be compared for equality conditions. Selections and text can not be used.

Error: The mix of variable or expression types can not be multiplied or divided

This message occurs in equations. Variables are type identified, so equation and assignments (x = 9) can be verified. In an equation only numerical values can be multiplied or divided. Booleans, Selections, text and events can not be compared.

Error: The Source Event- <variable name> can only be assigned <max number> times

The limits to Source Event assignments are the number of destinations assigned to a single Output event is limited. Generally this is three. For Selections it is one. The Waitfor Instruction temporarily assigns its Output Event Operands. This is subject to the assignment limitations. To use an Output Event in a waitfor instruction, there must be at least one free assignment.

Error: The Selection variable can only be assigned a Selection value.

The destination variable only accepts selection values. Selection values are fixed unquoted text. The selection text must exactly match the available selections of the Destination variable.

Error: This instance does not exist.

This variable is referencing an instance that has not been created in your application. For example “Index.9.vel” the instance 9 of index has not been defined.

Error: This is not a fully qualified variable.

To use a variable it must be fully defined. Some variables are global variables and only the name is defined. Other Variables require a name.name convention. Other variables require an instance number (index.1.vel). All the components identifying the variable must be available to qualify the variable.

Error: This variable cannot be assigned a selection value

The variable that you are attempting to define with a selection does not accept selection values.

Error: This variable type cannot be assigned a value

The Variable is of a type that does not accept any assignments. A Source Event variable is an example. You can not assign a value to an output event.

Error: The selection is not valid for this variable

The variable that you are attempting to define with a selection does accept selection values. However the selection you are attempting to use is not accepted by this variable.

Error: Trying to assign a selection variable with bad selection data

When checking to see if the selection goes with the destination variable, the source is not a defined selection for the destination variable.

Online Status Indicators

Global Where Am I Button

The Program View, when online and executing a program or sequence of programs, can display current program status. Pressing the Where Am I button on the PowerTools FM-3 toolbar creates a blue triangle that appears on the line of the program currently being executed.

The Global Where Am I can be used for diagnostics. When the user needs to know where in a complicated program the drive is or when the user wishes to follow the logical flow of the program.

Motion Status

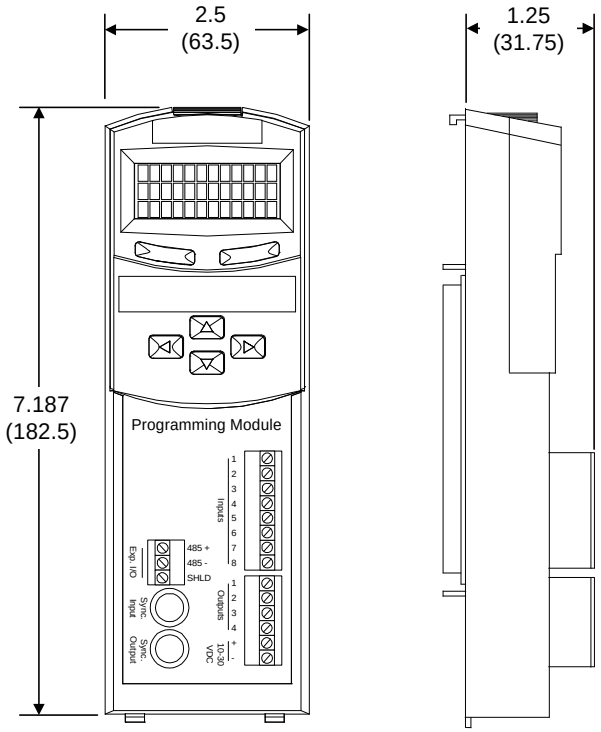
While the drive is online, the name of the program currently running or the motion type currently running will appear in the status bar at the bottom left corner of the PowerTools FM-3 screen.

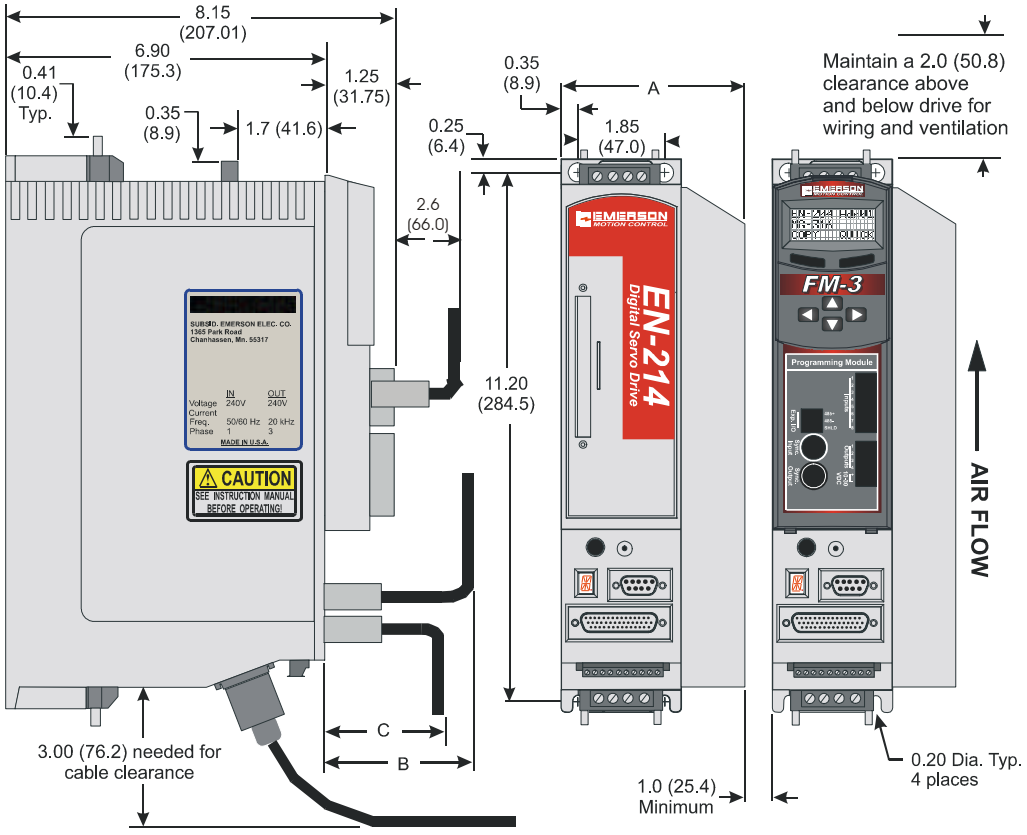
Specifications

Power consumption: 3W from E Series drive power supply.
I/O Supply Voltage: 10-30 VDC.

Function		Electrical Characteristics
Inputs	ON State Voltage	10-30 VDC
	ON State Current	2mA-6.5mA
	OFF State Voltage	0-3VDC
	OFF State Current	0-400µA
Outputs	Max. ON State Voltage	I/O Supply Voltage -1.5V
	Max. ON State Current	150mA
	Max. OFF State Current	100µA

Dimensions and Clearances



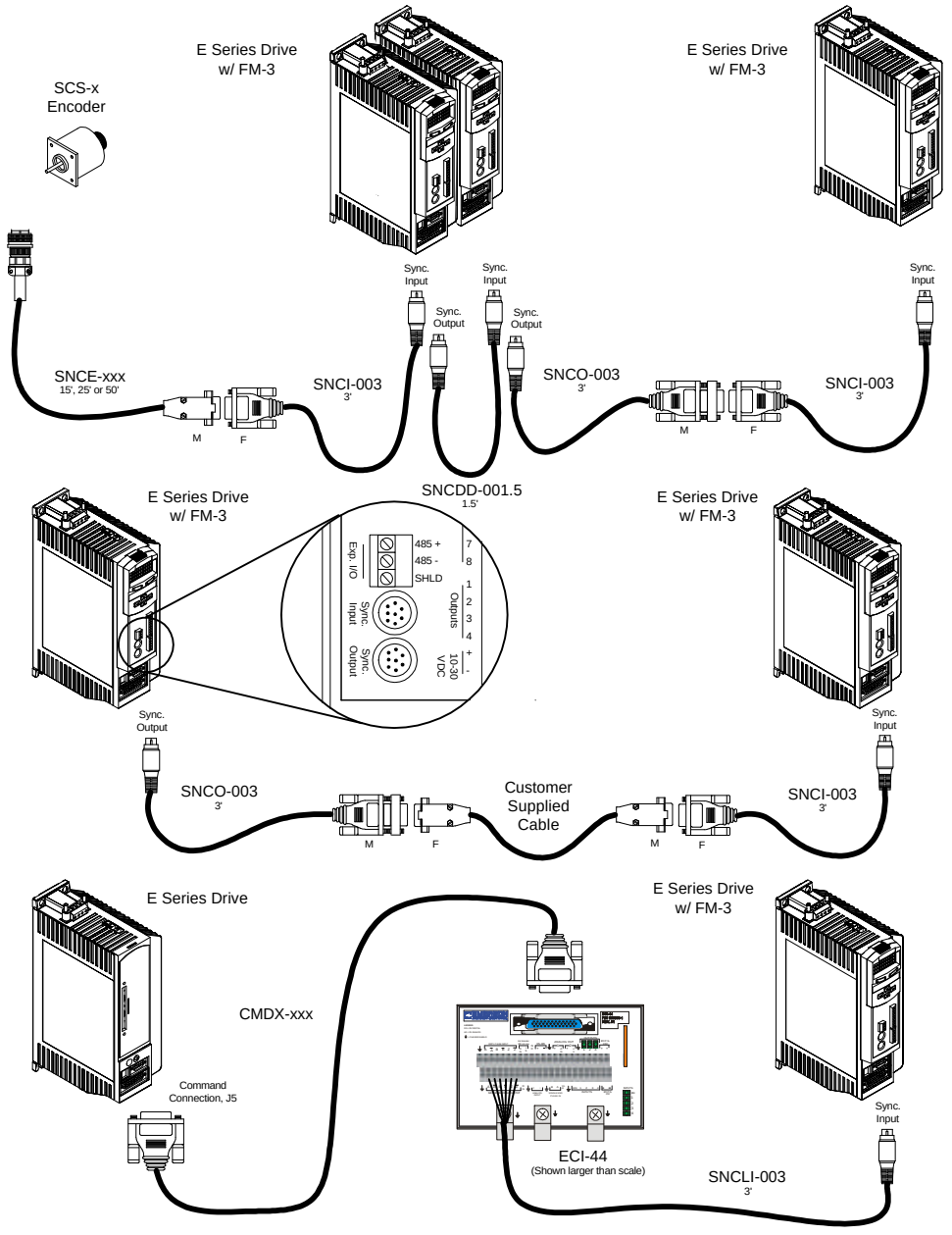


Model	A
EN-204	2.93 (74.4)
EN-208	3.43 (87.1)
EN-214	3.93 (99.8)

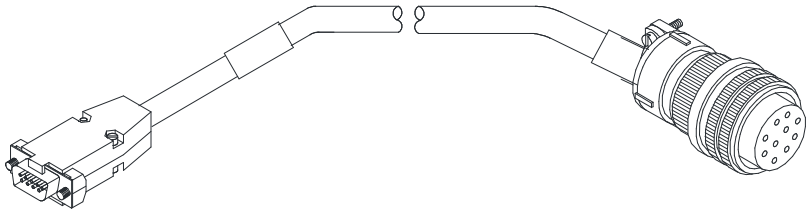
Model	B	C	
All Drive Models	TIA Cable CO485-XXX	CMDX CMDO	AX-CEN CDRO
	2.5 (63.5)	3.5 (88.9)	1.75 (44.5)

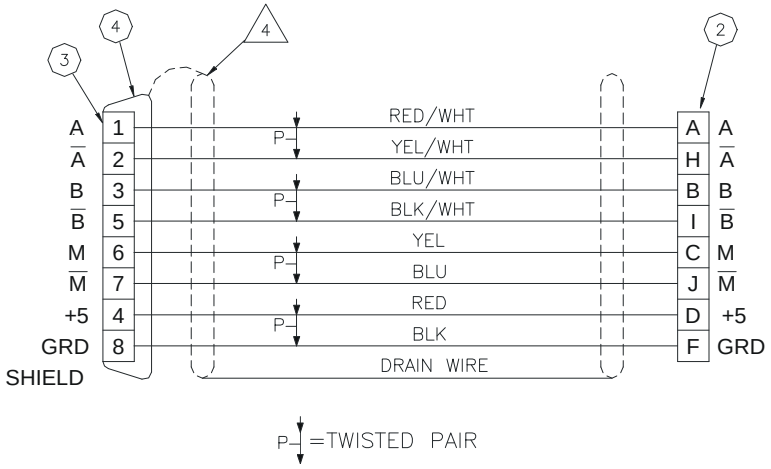
Dimensions in ( ) are millimeters

Cable Diagrams

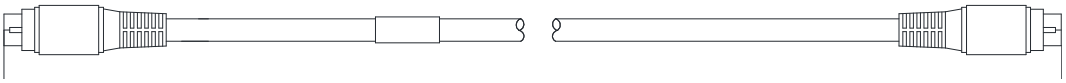


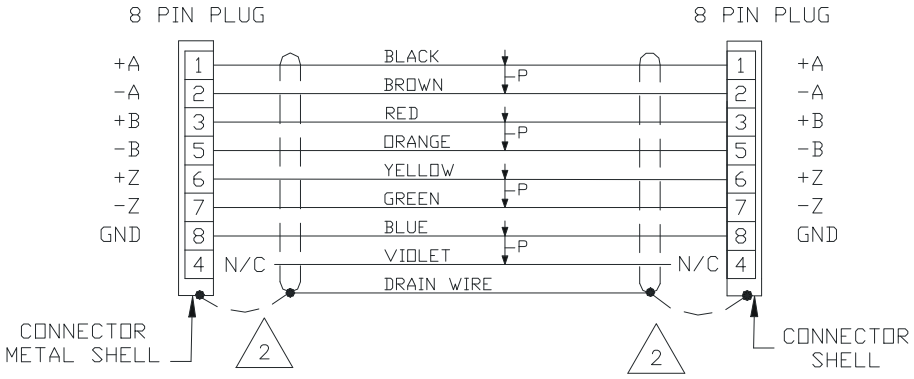
SNCE-XXX Cable



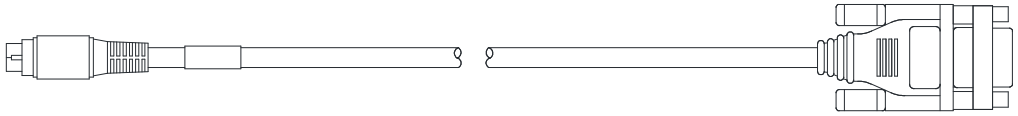


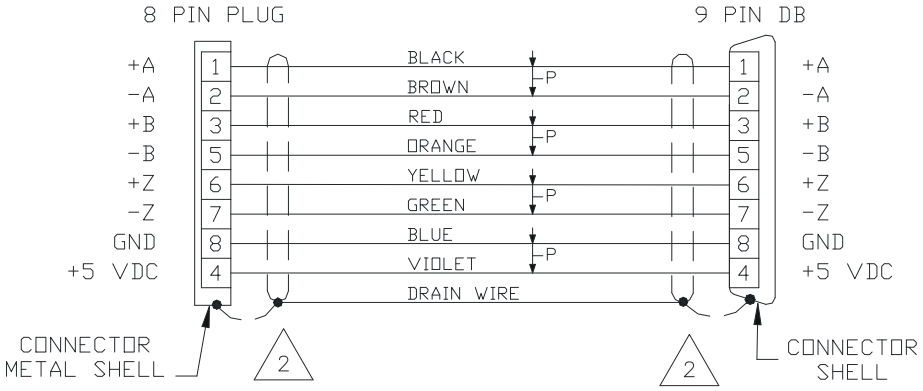
SNCDD-001.5 Cable



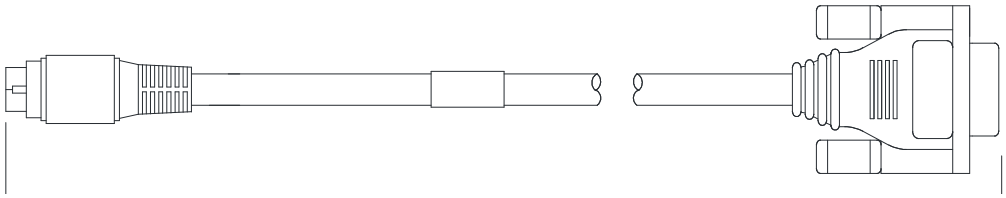


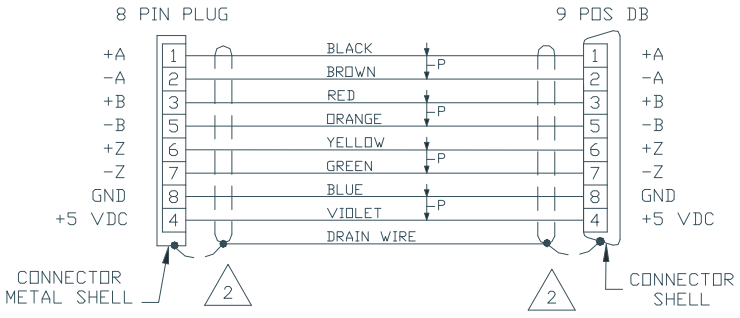
SNCO-003 Cable



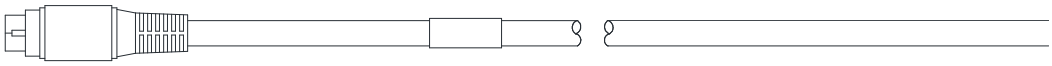


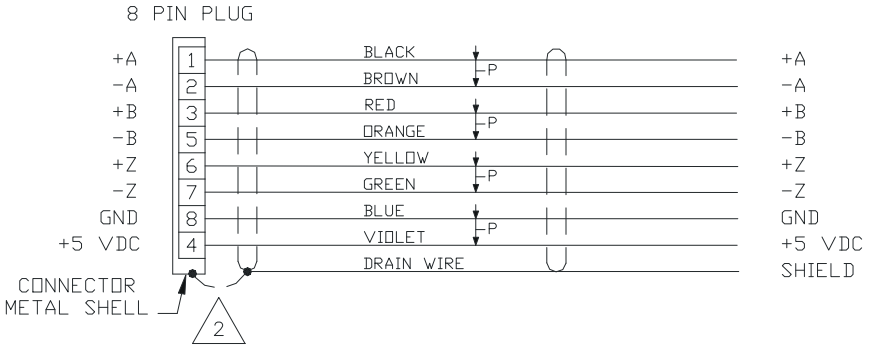
SNCI-003 Cable



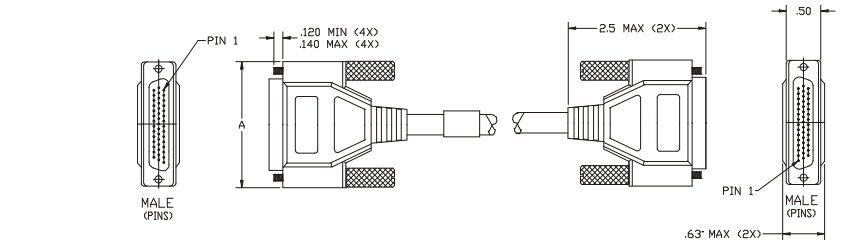


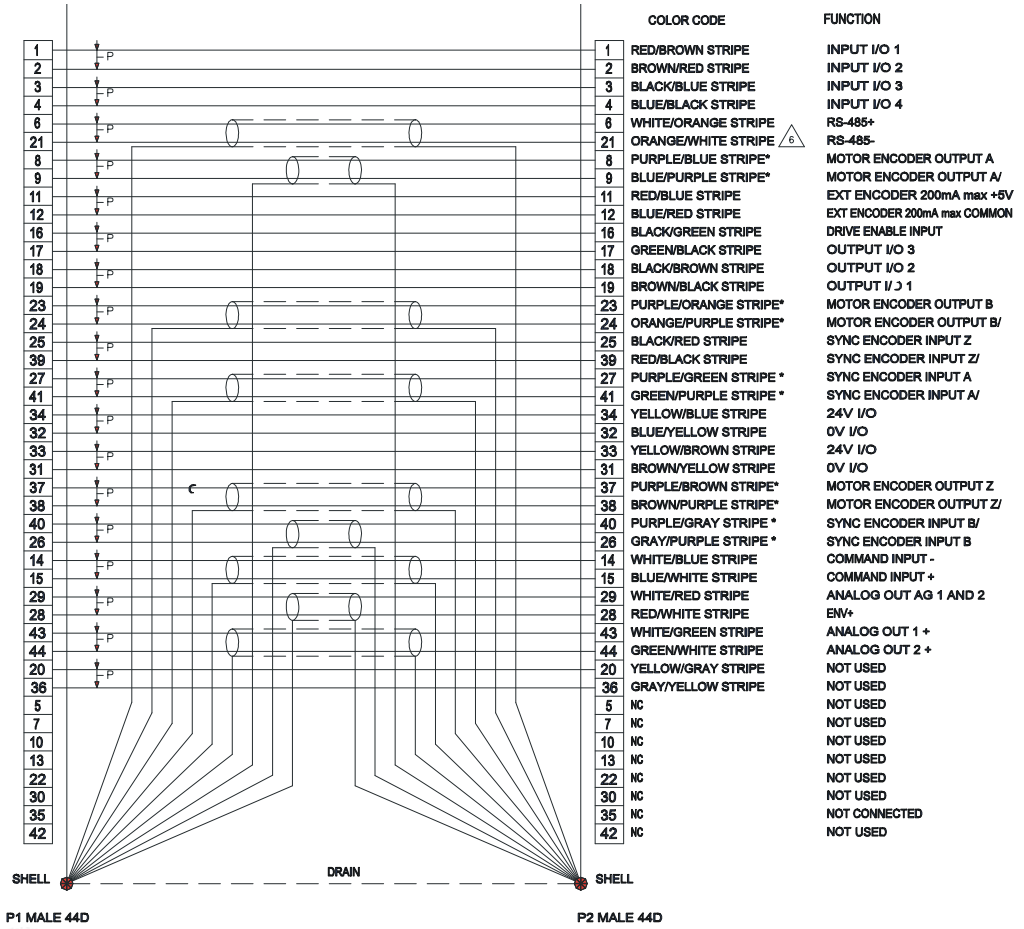
SNCLI-003 Cable





CMDX-XXX Cable



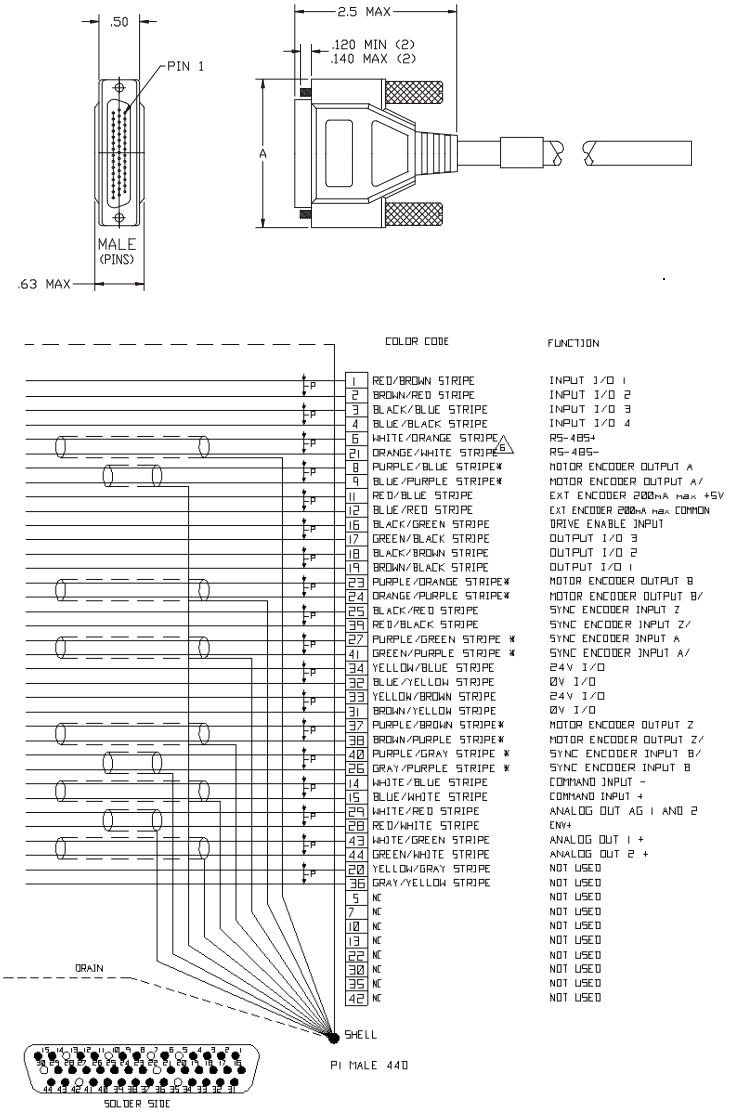


P1	P2	COLOR CODE	FUNCTION
1	1	RED/BROWN STRIPE	INPUT I/O 1
2	2	BROWN/RED STRIPE	INPUT I/O 2
3	3	BLACK/BLUE STRIPE	INPUT I/O 3
4	4	BLUE/BLACK STRIPE	INPUT I/O 4
6	6	WHITE/ORANGE STRIPE	RS-485+
21	21	ORANGE/WHITE STRIPE (6)	RS-485-
8	8	PURPLE/BLUE STRIPE*	MOTOR ENCODER OUTPUT A
9	9	BLUE/PURPLE STRIPE*	MOTOR ENCODER OUTPUT A/
11	11	RED/BLUE STRIPE	EXT ENCODER 200mA max +5V
12	12	BLUE/RED STRIPE	EXT ENCODER 200mA max COMMON
16	16	BLACK/GREEN STRIPE	DRIVE ENABLE INPUT
17	17	GREEN/BLACK STRIPE	OUTPUT I/O 3
18	18	BLACK/BROWN STRIPE	OUTPUT I/O 2
19	19	BROWN/BLACK STRIPE	OUTPUT I/O 1
23	23	PURPLE/ORANGE STRIPE*	MOTOR ENCODER OUTPUT B
24	24	ORANGE/PURPLE STRIPE*	MOTOR ENCODER OUTPUT B/
25	25	BLACK/RED STRIPE	SYNC ENCODER INPUT Z
39	39	RED/BLACK STRIPE	SYNC ENCODER INPUT Z/
27	27	PURPLE/GREEN STRIPE *	SYNC ENCODER INPUT A
41	41	GREEN/PURPLE STRIPE *	SYNC ENCODER INPUT A/
34	34	YELLOW/BLUE STRIPE	24V I/O
32	32	BLUE/YELLOW STRIPE	0V I/O
33	33	YELLOW/BROWN STRIPE	24V I/O
31	31	BROWN/YELLOW STRIPE	0V I/O
37	37	PURPLE/BROWN STRIPE*	MOTOR ENCODER OUTPUT Z
38	38	BROWN/PURPLE STRIPE*	MOTOR ENCODER OUTPUT Z/
40	40	PURPLE/GRAY STRIPE *	SYNC ENCODER INPUT B/
26	26	GRAY/PURPLE STRIPE *	SYNC ENCODER INPUT B
14	14	WHITE/BLUE STRIPE	COMMAND INPUT -
15	15	BLUE/WHITE STRIPE	COMMAND INPUT +
29	29	WHITE/RED STRIPE	ANALOG OUT AG 1 AND 2
28	28	RED/WHITE STRIPE	ENV+
43	43	WHITE/GREEN STRIPE	ANALOG OUT 1 +
44	44	GREEN/WHITE STRIPE	ANALOG OUT 2 +
20	20	YELLOW/GRAY STRIPE	NOT USED
36	36	GRAY/YELLOW STRIPE	NOT USED
5	5	NC	NOT USED
7	7	NC	NOT USED
10	10	NC	NOT USED
13	13	NC	NOT USED
22	22	NC	NOT USED
30	30	NC	NOT USED
35	35	NC	NOT CONNECTED
42	42	NC	NOT USED

Note

Some CMDX cables may have White/Yellow and Yellow/White wires in place of the White/Orange and Orange/White shown in the figure above (pins 6 and 21).

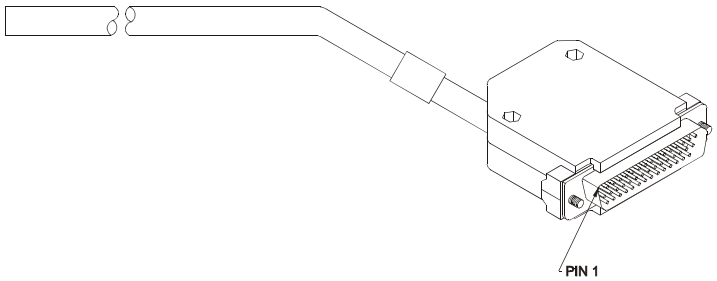
CMDO-XXX Cable

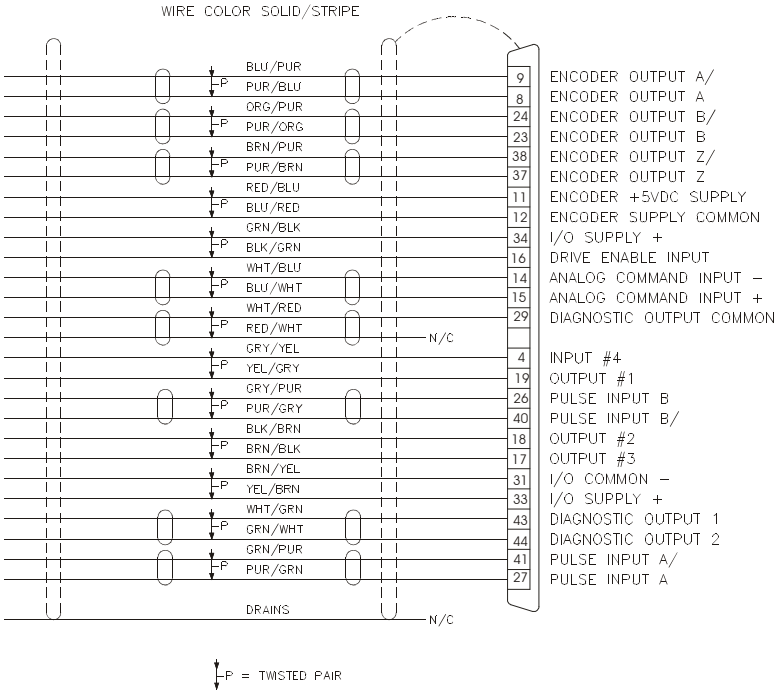

.50
PIN 1
MALE
(PINS)
.63 MAX
2.5 MAX
.120 MIN (2)
.140 MAX (2)
A
COLOR CODE
FUNCTION
1 RED/BROWN STRIPE — INPUT I/O 1
2 BROWN/RED STRIPE — INPUT I/O 2
3 BLACK/BLUE STRIPE — INPUT I/O 3
4 BLUE/BLACK STRIPE — INPUT I/O 4
6 WHITE/ORANGE STRIPE — RS-485+
21 ORANGE/WHITE STRIPE — RS-485-
8 PURPLE/BLUE STRIPE* — MOTOR ENCODER OUTPUT A
9 BLUE/PURPLE STRIPE* — MOTOR ENCODER OUTPUT A/
11 RED/BLUE STRIPE — EXT ENCODER 200mA max +5V
12 BLUE/RED STRIPE — EXT ENCODER 200mA max COMMON
16 BLACK/GREEN STRIPE — DRIVE ENABLE INPUT
17 GREEN/BLACK STRIPE — OUTPUT I/O 3
18 BLACK/BROWN STRIPE — OUTPUT I/O 2
19 BROWN/BLACK STRIPE — OUTPUT I/O 1
23 PURPLE/ORANGE STRIPE* — MOTOR ENCODER OUTPUT B
24 ORANGE/PURPLE STRIPE* — MOTOR ENCODER OUTPUT B/
25 BLACK/RED STRIPE — SYNC ENCODER INPUT Z
39 RED/BLACK STRIPE — SYNC ENCODER INPUT Z/
27 PURPLE/GREEN STRIPE * — SYNC ENCODER INPUT A
41 GREEN/PURPLE STRIPE * — SYNC ENCODER INPUT A/
34 YELLOW/BLUE STRIPE — 24V I/O
32 BLUE/YELLOW STRIPE — 0V I/O
33 YELLOW/BROWN STRIPE — 24V I/O
31 BROWN/YELLOW STRIPE — 0V I/O
37 PURPLE/BROWN STRIPE* — MOTOR ENCODER OUTPUT Z
38 BROWN/PURPLE STRIPE* — MOTOR ENCODER OUTPUT Z/
40 PURPLE/GRAY STRIPE * — SYNC ENCODER INPUT B/
26 GRAY/PURPLE STRIPE * — SYNC ENCODER INPUT B
14 WHITE/BLUE STRIPE — COMMAND INPUT -
15 BLUE/WHITE STRIPE — COMMAND INPUT +
29 WHITE/RED STRIPE — ANALOG OUT AG 1 AND 2
28 RED/WHITE STRIPE — ENV+
43 WHITE/GREEN STRIPE — ANALOG OUT 1 +
44 GREEN/WHITE STRIPE — ANALOG OUT 2 +
20 YELLOW/GRAY STRIPE — NOT USED
36 GRAY/YELLOW STRIPE — NOT USED
5 NC — NOT USED
7 NC — NOT USED
10 NC — NOT USED
13 NC — NOT USED
22 NC — NOT USED
30 NC — NOT USED
35 NC — NOT USED
42 NC — NOT USED
6
DRAIN
SHELL
P1 MALE 44D
SOLDER SIDE

Note

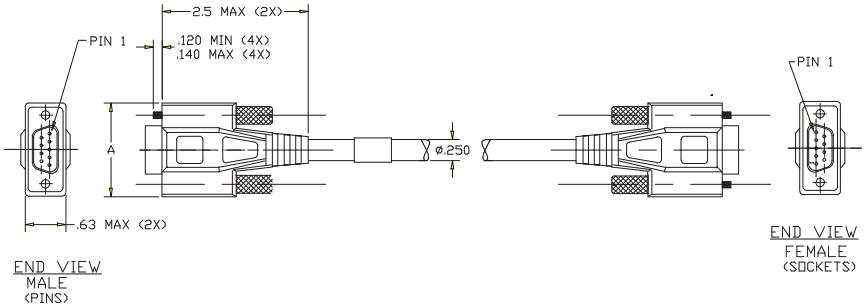
Some CMDO cables may have White/Yellow and Yellow/White wires in place of the White/Orange and Orange/White shown in the figure above (pins 6 and 21).

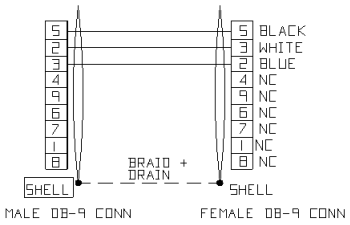
CDRO-XXX Cable



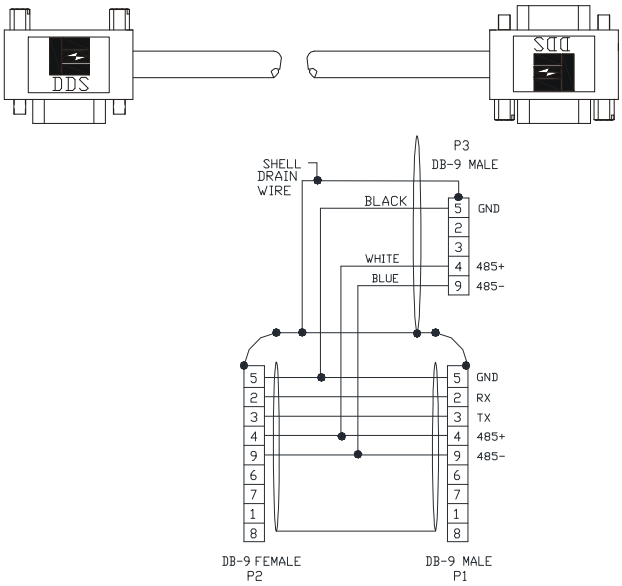


TIA-XXX Cable

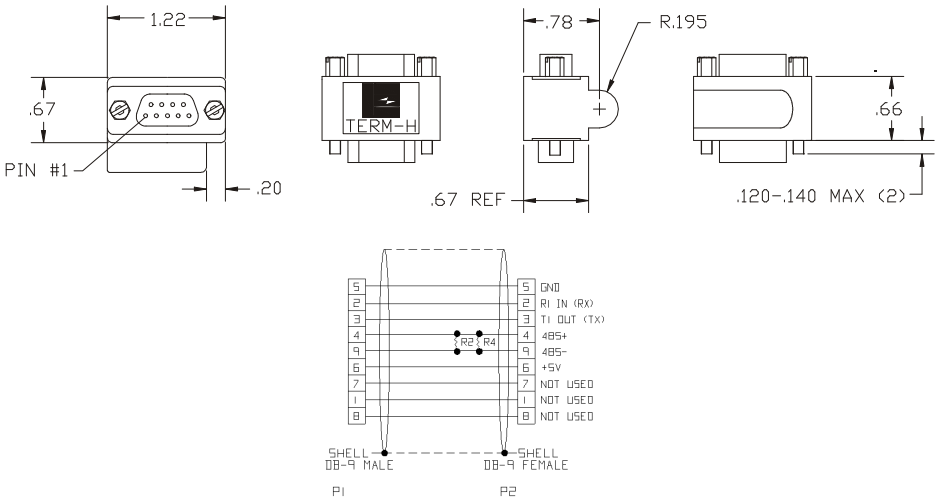




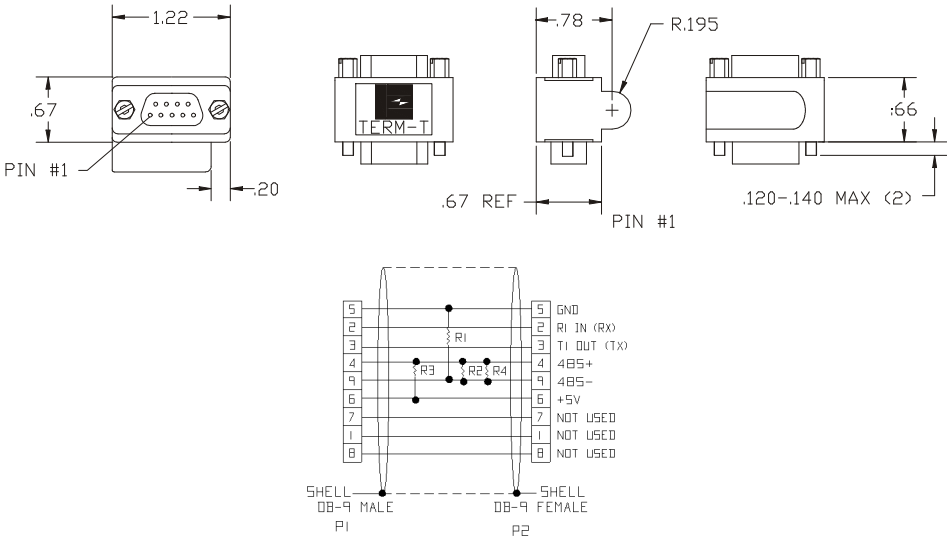
DDS-XXX Cable



TERM-H (Head) Terminator



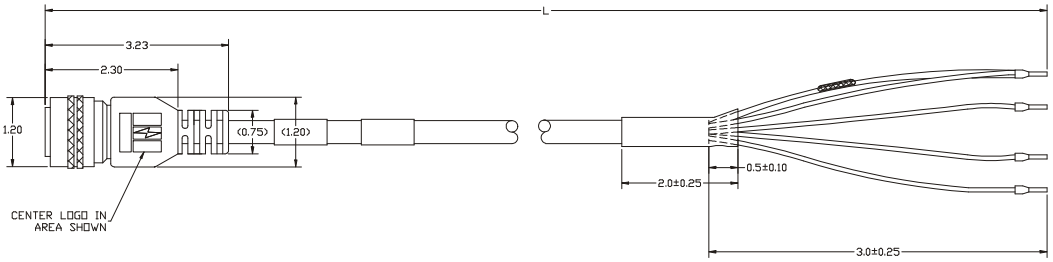
TERM-T (Tail) Terminator

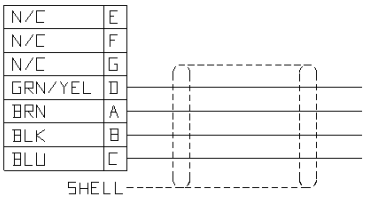


Note

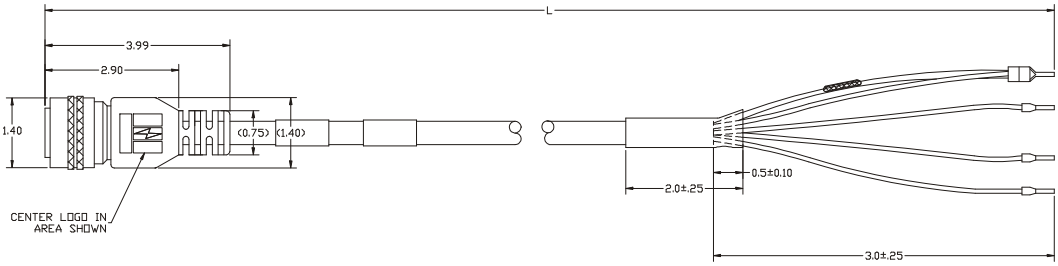
See the "Multi-drop Communications" section for resistor values.

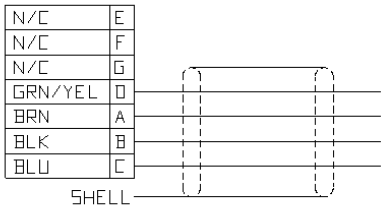
CMDS-XXX Cable



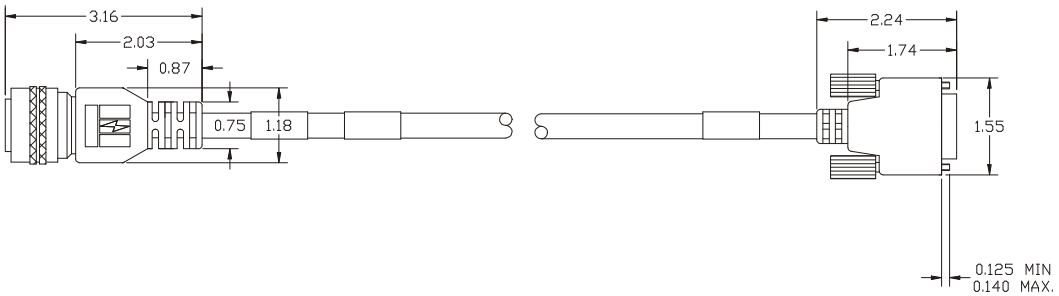


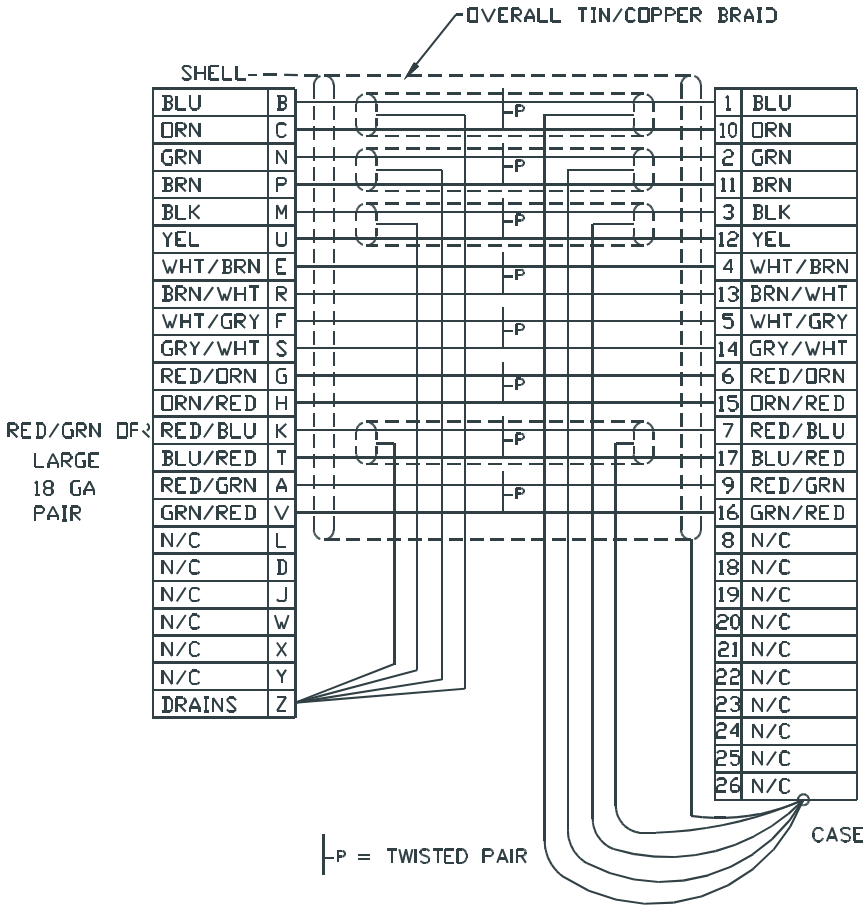
CMMS-XXX Cable



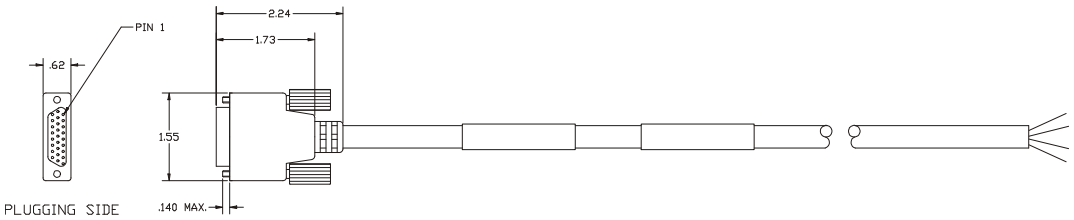


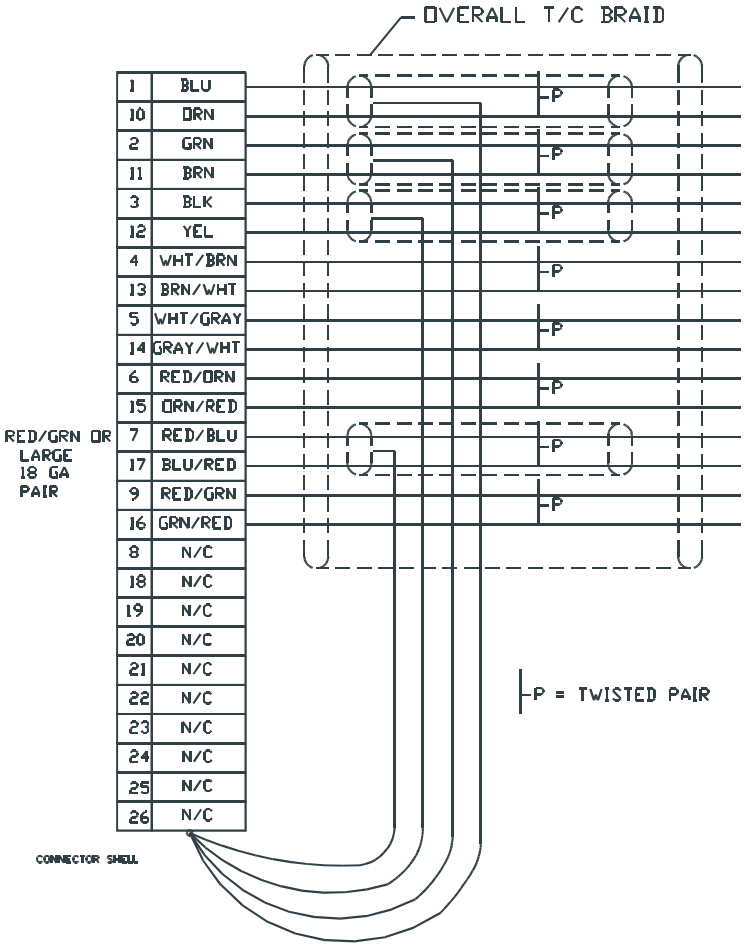
CFCS-XXX Cable



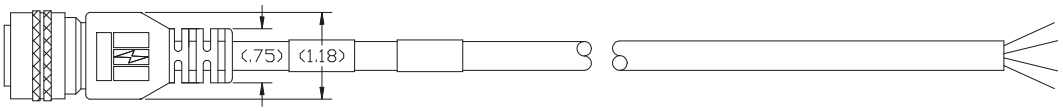


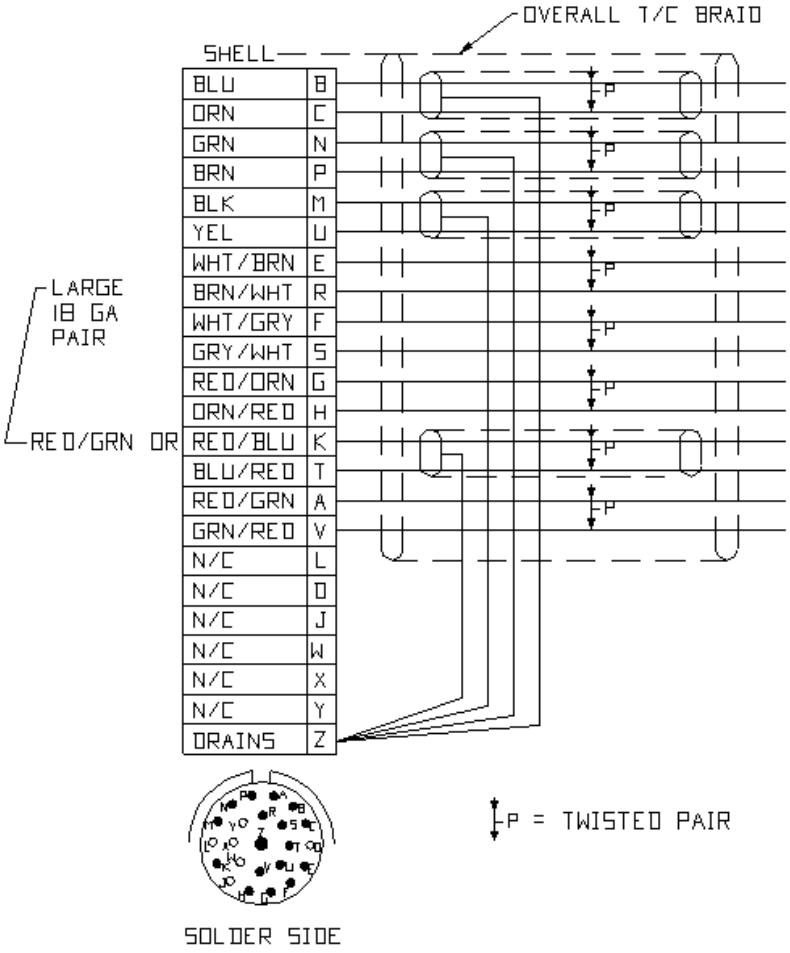
CFCO-XXX Cable





CFOS-XXX Cable





Vendor Contact Information

Schaffner (AC Line Filters)
(800) 367-5566 or (201) 379-7778 www.schaffner.com

Cooper Industries, Inc.
Crouse-Hinds Division (Cable Shield Grommets)
(315) 477-5531 www.crouse-hinds.com

Bussman
P.O. Box 14460
St. Louis, MO. 63178-4460
(314) 394-3877 www.bussman.com

Littlefuse
800 E. Northwest Hwy
Des Plaines, IL. 60016
(847) 824-0400 www.littlefuse.com

Wickmann USA
4100 Shirlel Dr.
Atlanta, GA. 30336
(404) 699-7820 www.wickmann.com

Corcom
844 E. Rockland Road
Libertyville, IL 60048
(847) 680-7444 www.corcom.com

Glossary

µs

Microsecond, which is 0.000001 seconds.

A

Amps.

Amplifier

Servo Drive.

ARMS

Amps Root Mean Squared (RMS).

Axis

The full system to control in a single motor shaft. A single FM-3 with EN Drive can denote an axis.

AWG

American Wire Gauge.

Baud Rate

The number of binary bits transmitted per second on a serial communications link such as RS-232. (1 character is usually 10 bits.)

Check Box

In a dialog box, a check box is a small box that the user can turn “On” or “Off” with the mouse. When “On” it displays an X in a square; when “Off” the square is blank. Unlike option (radio) buttons, check boxes do not affect each other; any check box can be “On” or “Off” independently of all the others.

Complex Motion

A string of multiple motion commands and logical instructions that form a repeatable operation. For the FM-3, the configuration file defines comlpex motion by setups, functional assignments and programs.

Compound Motion

The combination of indexes in a row in which the deceleration ramp of the first index goes to the velocity of the secondary index. The first index must be initiated within a program (Index.#.CompoundInitiate).

Configuration

The user-created application. It can be saved as a disk file or downloaded to configure the FM-3. It uncludes all the user-defined setup, assignments and programs.

CRC

Cyclical Redundancy Check, the data transfer error checking mechanism.

Destination

A function (i.e., Stop, Preset) that may be assigned to an input line. In FM-3, the input function is connected to the action through click and drag operations in Emerson Motion Control PowerTools® Software on the Assignment View.

Dialog Box

A dialog box is a window that appears in order to collect information from the user. When the user has filled in the necessary information, the dialog box disappears.

DIN Rail

Deutsche Industrie Norm Rail

DLL

In Microsoft® Windows®, a Dynamic Link Library contains a library of machine-language procedures that can be linked to programs as needed at run time.

Downloading

The transfer of a complete set of parameters from an FM to a drive.

Drive

Servo drive or amplifier.

EEPROM

An EEPROM chip is an Electrically Erasable Programmable Read-Only Memory; that is, its contents can be both recorded and erased by electrical signals, but they do not go blank when power is removed.

EMC

Electromagnetic Compatibility. The relative immunity of a drive to the effects of electromagnetic fields.

EMI - Electro-Magnetic Interference

EMI is noise which, when coupled into sensitive electronic circuits, may cause problems.

Firmware

The term firmware refers to software (i.e., computer programs) that are stored in some fixed form, such as read-only memory (ROM).

Flash

Another type of EEPROM.

Flash File

In the FM-3, this file loads the firmware into the drive and function module. Flash files can field upgrade the firmware.

FM

Function Module - device which is attached to the front of the drive to provide additional functionality.

Global Where Am I

Emerson Motion Control PowerTools feature that indicates which line of which user program is executing.

Home Routine

The home provides motion in applications in which the axis must precisely align with some part of a machine.

Hysteresis

For a system with an analog input, the output tends to maintain its current value until the input level changes past the point that set the current output value. The difference in response of a system to an increasing input signal versus a decreasing input signal.

I/O

Input/Output. The reception and transmission of information between control devices. In modern control systems, I/O has two distinct forms: switches, relays, etc., which are in either an on or off state, or analog signals that are continuous in nature generally depicting values for speed, temperature, flow, etc.

Index

An index is a complete motion sequence (defined motion profile) that moves the motor a specific incremental distance or to an absolute position.

Inertia

The property of an object to resist changes in rotary velocity unless acted upon by an outside force. Higher inertia objects require larger torque to accelerate and decelerate. Inertia is dependent upon the mass and shape of the object.

Input Function

See destination. A function (i.e., Stop, Preset) that may be assigned to an input line. In FM-3, the input function is connected to the action through click and drag operations in Emerson Motion Control PowerTools Software on the Assignment View.

Input Line

The terminals of a device or circuit to which energy is applied.

Jog

A jog produces rotation of the motor at controlled velocities in a positive or negative direction.

Least Significant Bit

The bit in a binary number that is the least important or having the least weight.

LED

Light Emitting Diode used on the front display of drives and function modules.

List Box

In a dialog box, a list box is an area in which the user can choose among a list of items, such as files, directories, printers or the like.

mA

Milliamp, which is 1/1000th of an Ampere.

MB

Mega-byte.

MODBUS

Communication Protocol by Modicon. The E Series follows the Modbus specification outlined in the Modicon Modbus Protocol Reference Manual, PI-MBNS-300 Revision G, November 1994.

Module

Function Module

Most Significant Bit

The bit in a binary number that is the most important or that has the most weight.

ms

Millisecond, which is 1/1000th of a second.

NVM

Non-Volatile Memory. NVM stores specifically defined variables as the variables dynamically change. It is used to store changes through a power loss.

NTC

Negative Temperature Resistor

Option Button

See Radio Button.

Opto-isolated

A method of sending a signal from one piece of equipment to another without the usual requirement of common ground potentials. The signal is transmitted optically with a light source (usually a Light Emitting Diode) and a light sensor (usually a photosensitive transistor). These optical components provide electrical isolation.

Output Function

See source. The terminals at which energy is taken from a circuit or device.

Output Line

The actual transistor or relay controlled output signal.

Parameters

User read only or read/write parameters that indicate and control the drive operation. These variables generally hold numerical data defined in the Setup Views.

PC

Personal Computer.

PE

Protective Earth.

PID

Proportional-Integral-Derivative. An acronym that describes the compensation structure that can be used in many closed-loop systems.

PLC

Programmable Logic Controller. Also known as a programmable controller, these devices are used for machine control and sequencing.

Emerson Motion Control PowerTools-Base, -FM and -FM-3

Windows based software to interface with the Epsilon drives, E Series drives and Function Modules.

Radio Button

Also known as the Option Button. In a dialog box, radio buttons are small circles only one of which can be chosen at a time. The chosen button is black and the others are white. Choosing any button with the mouse causes all the other buttons in the set to be cleared.

RAM

RAM is an acronym for Random-Access Memory, which is a memory device whereby any location in memory can be found, on average, as quickly as any other location. Commonly refers to Read-Write memory, as opposed to Read-Only Memory (ROM, EPROM, EEPROM, Flash). RAM is considered volatile, because its contents are lost during a power loss.

RMS

Root Mean Squared. For an intermittent duty cycle application, the RMS is equal to the value of steady state current which would produce the equivalent heating over a long period of time.

ROM

ROM is an acronym for Read-Only Memory. A ROM contains computer instructions that do not need to be changed, such as permanent parts of the operating system.

RPM

Revolutions Per Minute.

Serial Port

A digital data communications port configured with a minimum number of signal lines. This is achieved by passing binary information signals as a time series of 1’s and Ø’s on a single line.

Source

The terminals at which energy is taken from a circuit or device.

Travel Limit

The distance that is limited by either a travel limit switch or the software.

Torque

The moment of force, a measure of its tendency to produce torsion and rotation about an axis.

Uploading

The transfer of a complete set of parameters from a drive to an FM.

User Units

Ability of program to allow user to specify which type of units will measure and specify motion and time.

VAC

Volts, Alternating Current.

Variable

A labeled value that encompasses numeric boolean, input function, and output functions.

VDC

Volts, Direct Current.

Velocity

The rate of change in position in a given direction during a certain time interval.

View

Portion of screen within frame.

Windows®, Microsoft®

Microsoft Windows is an operating system that provides a graphical user interface, extended memory and multi-tasking. The screen is divided into windows and the user uses a mouse to start programs and make menu choices.

Index

Symbols

+/- Limit, 174

A

All "On", 174
Analog Output, 177

B

Brake Operation and Wiring, 176

C

Cable Diagrams, 191
CDRO-XXX Cable, 197
CFCO-XXX Cable, 202
CFCS-XXX Cable, 201
CFOS-XXX Cable, 203
CMDO-XXX Cable, 196
CMDS-XXX Cable, 200
CMDX-XXX Cable, 195
CMMS-XXX Cable, 200

D

DDS-XXX Cable, 198
Diagnostic Display, 169
Digital Inputs and Outputs, 61
Drive Faults, 180

E

Encoder Line Fault, 174
Encoder State, 173
Error Messages, 180

F

Fault Codes, 170
Fault Descriptions, 172
Firmware Checksum, 172
Following Error Fault, 174

H

High DC Bus Fault, 173

I

Installation Notes, 129
Introduction, 1
Invalid Configuration, 172, 173

L

Low DC Bus Fault, 173

M

Modbus Communications, 130
Motor Over Temperature Fault, 174

N

Non-volatile Memory Invalid, 172

O

Over Speed Fault, 174

P

Power Stage Fault, 173
Power-Up Self-Test Failure, 172
Programming Error Messages, 182

R

Registration Index, 20
RMS Shunt Power Fault, 174
Rotary + and Rotary - Indexes, 21
Rotary Indexes, 21

S

Safety Considerations, vii
Safety of Machinery, vii
Safety Precautions, vii
Setup, 25
Setup, Commissioning and Maintenance, vii
status codes
    decimal point, 169
    Ready, 169
    Ready to Run, 169

T

TERM-H (Head) Terminator, 199
TERM-T (Tail) Terminator, 199
TIA-XXX Cable, 198

W

Watchdog Timer, 172

Since 1979, EMERSON Motion Control, a subsidiary of the Emerson Electric Company, has been a leader in the development and manufacturing of motion control equipment and software. EMERSON Motion Control continues to lead the industry in supplying motion control devices designed to improve production efficiency.

For more information about EMERSON Motion Control products and services, call 1.800.39-SERVO or contact our website at www.emersondrivesolutions.com.

EMERSON Motion Control
Subsidiary of Emerson Electric Co.
12005 Technology Drive
Eden Prairie, Minnesota 55344
U.S.A.

Sales: 952.995.8000 or 1.800.39-SERVO
Service: 952.995.8033
Fax: 952.995.8011

Printed in U.S.A.

EMERSON
MOTION CONTROL